



Ministério da
Ciência e Tecnologia



INPE-15672-TDI/1446

ANÁLISE DO DETERMINISMO NA COMUNICAÇÃO POR REDE EM SISTEMAS COMPUTACIONAIS EMBARCADOS EM VEÍCULOS AEROESPACIAIS

Sérgio Duarte Penna

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelos Drs. Marcelo Lopes de Oliveira e Souza e Atair Rios Neto, aprovada em 12 de dezembro de 2008.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m18@80/2008/11.26.12.38>>

INPE
São José dos Campos
2009

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:

Presidente:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)

Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
Ciência e Tecnologia



INPE-15672-TDI/1446

ANÁLISE DO DETERMINISMO NA COMUNICAÇÃO POR REDE EM SISTEMAS COMPUTACIONAIS EMBARCADOS EM VEÍCULOS AEROESPACIAIS

Sérgio Duarte Penna

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelos Drs. Marcelo Lopes de Oliveira e Souza e Atair Rios Neto, aprovada em 12 de dezembro de 2008.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m18@80/2008/11.26.12.38>>

INPE
São José dos Campos
2009

Dados Internacionais de Catalogação na Publicação (CIP)

P381a Penna, Sérgio Duarte.

Análise do determinismo na comunicação por rede em sistemas computacionais embarcados em veículos aeroespaciais / Sérgio Duarte Penna. – São José dos Campos: INPE, 2009.

105p. ; (INPE-15672-TDI/1446)

Dissertação (Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2008.

1. Sistemas embarcados. 2. Veículos aeroespaciais. 3. Rede de comunicação. 4. Determinismo. I.Título.

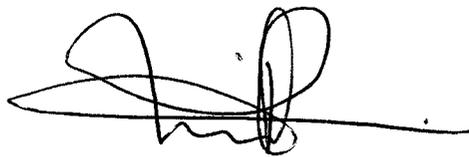
CDU 629.7.06

Copyright © 2009 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, microfílmico, reprográfico ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2009 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

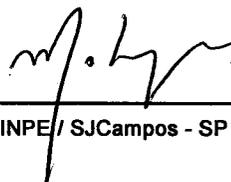
Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Mestre em
Engenharia e Tecnologia Espaciais/Mecânica
Espacial e Controle

Dr. Paulo Giácomo Milani



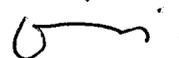
Presidente / INPE / SJCampos - SP

Dr. Marcelo Lopes de Oliveira e Souza



Orientador(a) / INPE / SJCampos - SP

Dr. Atair Rios Neto



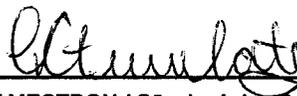
Orientador(a) / EMBRAER / SJCampos - SP

Dr. Celso Braga de Mendonça



Convidado(a) / EMBRAER / SJCampos - SP

Dr. Gilberto da Cunha Trivelato



Convidado(a) / MECTRON / São José dos Campos - SP

Aluno (a): Sergio Duarte Penna

São José dos Campos, 12 de dezembro de 2008

“Não são nossas habilidades que fazem o que somos, são nossas escolhas”.

Alvo Dumbledore

Aos meus orientadores e aos meus colegas da Pós-Graduação do INPE e da Engenharia da EMBRAER.

AGRADECIMENTOS

Agradeço aos meus orientadores e aos Engenheiros Hermínio Duque Lustosa, Celso Braga de Mendonça e Antônio Magno Lima Espeschit pelo apoio na construção e revisão deste texto. Agradeço também à Professora Doutora Cristina Duarte Murta pelo apoio na compreensão dos princípios do “Network Calculus”, ao Engenheiro Cláudio Alexandre da Costa Dias pelo apoio no uso do “Solver” do Microsoft Excel[®], ao Engenheiro Augusto Marasca de Conto pela sua contribuição na compreensão do algoritmo Welch-Lynch de sincronização de relógios em sistemas distribuídos e à Engenheira Hialina Lima do Nascimento pela sua contribuição no cálculo de ciclos TDMA.

RESUMO

Sistemas computacionais embarcados em veículos aeroespaciais têm assumido um caráter distribuído na tentativa de resolver problemas como reconfigurabilidade, escalabilidade, manutenibilidade, comunalidade e longevidade de componentes e subsistemas. Em um sistema de computação distribuída, processos e tarefas são executados por computadores similares interligados por um canal de comunicação. Por outro lado, sistemas distribuídos apresentam desafios de projeto bem maiores que os tradicionais sistemas ditos “monolíticos”, onde todas as funções são executadas por um único computador combinando hardware e software. Dentre eles, o “determinismo” (a univocidade causal nas ocorrências de eventos), a “previsibilidade” (a univocidade temporal nas ocorrências de eventos), a “causalidade” (a ordenação temporal nas ocorrências de eventos do tipo causa e efeito) e o “sincronismo” (a simultaneidade nas ocorrências de eventos) podem ser apontados como grandes desafios ao projeto de sistemas. Este trabalho tem por objetivo a análise do determinismo na comunicação por rede em sistemas computacionais embarcados em veículos aeroespaciais. Um caso de estudo representado por um modelo simples de um corretor de trajetória é analisado exercitando técnicas disponíveis, em particular usando fundamentos do “Network Calculus” e a multiplexação por divisão do tempo, buscando assegurar um caráter determinístico à comunicação de dados. Os resultados obtidos são comparados e é mostrado que as análises são suficientes para determinar parâmetros essenciais para uma escolha correta do meio físico e da estratégia apropriada para transmissão de dados ainda em fase de projeto. Finalmente, as limitações encontradas nas análises feitas são discutidas e novos estudos são sugeridos.

ANALYSIS OF DETERMINISM IN NETWORK COMMUNICATIONS IN COMPUTER SYSTEMS ONBOARD AEROSPACE VEHICLES

ABSTRACT

Computer systems onboard aerospace vehicles have become more and more distributed in an attempt to solve "real-life" problems such as reconfigurability, scalability, maintainability, commonality and longevity of components and subsystems. On the other hand, distributed systems pose a much bigger challenge in system design than traditional, "monolithic" systems, whereby functions are performed by a single component combining hardware and software. Among them, the "determinism" (the causal univocity in the occurrences of events), "predictability" (the temporal univocity in the occurrences of events), "causality" (temporal ordination in the occurrences of cause and effect events) and "synchronism" (simultaneousness in the occurrences of events) can be pointed out as major challenges in system design. This work intends to analyze determinism in network communications in computer systems onboard aerospace vehicles. A simple model of a trajectory corrector is introduced and analyzed under a few available techniques, such as those derived from "Network Calculus" theory and from Time-Division Multiplexing, aiming to ensure determinism in data communication. Results obtained are compared and it is shown that the analysis performed are sufficient for obtaining essential parameters for a correct choice of the physical medium and data transmission strategy during a design phase. Finally, the limitations found are discussed and further studies are suggested.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO	21
1.1 Contexto e Motivação do Trabalho	21
1.2 Objetivo do Trabalho.....	23
1.3 Organização do Trabalho.....	24
2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA	27
2.1 Conceitos Básicos	27
2.2 Fundamentos de “Network Calculus”	30
2.3 Revisão de Topologias e Elementos de Rede	41
3 AVALIAÇÃO DO USO DA ABORDAGEM “RATE MONOTONIC” NA COMUNICAÇÃO DE DADOS POR REDE	47
4 FORMULAÇÃO DO PROBLEMA E DESCRIÇÃO DO MODELO DO CORRETOR DE TRAJETÓRIA	55
4.1 Formulação do Problema.....	55
4.2 Descrição do Modelo do “Corretor de Trajetória”	56
5 ANÁLISE DO DETERMINISMO NA COMUNICAÇÃO PONTO-A-PONTO EM UMA IMPLEMENTAÇÃO FEDERADA DO CORRETOR DE TRAJETÓRIA	59
5.1 Introdução	59
5.2 Análise do Determinismo na Comunicação Ponto-a-Ponto	61
6 ANÁLISES DO DETERMINISMO NA COMUNICAÇÃO POR REDE EM IMPLEMENTAÇÕES INTEGRADAS DO CORRETOR DE TRAJETÓRIA	65
6.1 Introdução	65
6.2 Análise do Determinismo em uma Topologia em “Estrela” usando um “Switch”	65
6.3 Reanálise Usando Filas de Alta e Baixa Prioridade	71
6.4 Análise do Determinismo em uma Topologia em “Barramento” usando “Time-Division Multiple Access” (TDMA)	75
6.5 Algoritmo Welch-Lynch de Correção de Relógios em Sistemas Distribuídos	84
6.6 Revisão da Análise do Determinismo em uma Topologia em “Barramento” Usando um Modelo “Produtor-Consumidor”	87
7 CONCLUSÕES E PROPOSTAS PARA ESTUDOS FUTUROS	99
7.1 Conclusões	99
7.2 Propostas para Possíveis Trabalhos Futuros	101
REFERÊNCIAS BIBLIOGRÁFICAS	103

LISTA DE FIGURAS

2.1 – Função cumulativa em “Network Calculus”.....	30
2.2 – Curva de Chegada em “Network Calculus”.....	31
2.3 – Curva de Serviço em “Network Calculus”.....	32
2.4 – Limite superior do tráfego retido em um sistema.....	33
2.5 – Atraso máximo em um fluxo de entrada.....	34
2.6 – Função “Affine” (à esquerda) e função “Rate-Latency” (à direita).	35
2.7 – Convolução de funções “Affine” $\alpha_{r,b}$ e “Rate-Latency” $\beta_{R,T}$	36
2.8 – Deconvolução de funções “Affine” $\alpha_{r,b}$ e “Rate-Latency” $\beta_{R,T}$	36
2.9 – Os três limites em funções “Affine” $\alpha_{r,t}$ e “Rate-Latency” $\beta_{R,T}$	37
2.10 – Evolução do tráfego retido usando “rate functions”.....	38
2.11 – Fluxo de Saída usando “rate functions”.....	39
2.12 – Limite do fluxo de saída para um caso $R > r$	40
2.13 – Limite do fluxo de saída para um caso $R < r$	40
2.14 – Topologias de Rede: (a) “barramento, (b) “anel”, (c) “estrela”.....	42
2.15 – “Switches” Matrix® N-Series da Enterasys Networks, Inc.....	44
2.16 – Elementos lógicos de um “switch” Ethernet.....	45
3.1 – Gráfico de Gantt para o conjunto de tarefas da Tabela 5.1.....	49
3.2 – Três unidades compartilhando um canal de comunicação.....	51
3.3 – Gráfico de Gantt para o conjunto de mensagens da Tabela 3.2.	52
5.1 – Corretor de trajetória em implementação “federada”.....	59
5.2 – Transmissão cíclica usando “minor frames e “major frames”.....	64
6.1 – Corretor de trajetória usando topologia “estrela”.....	66
6.2 – Modelo da porta física do “switch” ligada à FCM.....	67
6.3 – Corretor de trajetória usando topologia em “barramento”.....	75
6.4 – Formato do ciclo TDMA para o corretor de trajetória.	80
6.5 – Sequência de transmissão TDMA para o corretor de trajetória.....	81
6.6 – Exemplo simplificado de aplicação do algoritmo Welch-Lynch.	85
6.7 – Nova implementação do corretor de trajetória.....	88
6.8 – Produção e consumo de dados no corretor de trajetória.....	90
6.9 – Uma “rodada” (ou “round”) de produção e consumo de dados.	91
6.10 – Um ciclo completo de produção e consumo de dados.	93
6.11 – Tempos de processamento na primeira rodada do ciclo.	94
6.12 – Um ciclo alternativo de produção e consumo de dados.	95
6.13 – Outro ciclo alternativo de produção e consumo de dados.....	96

LISTA DE TABELAS

3.1 – Tabela de tarefas para teste de agendabilidade.	48
3.2 – Tabela de mensagens para teste de transmissibilidade.....	51
5.1 – Tabela de mensagens para o corretor de trajetória.....	61
5.2 – Tabela de mensagens para a unidade FCM.	62
6.1 – Tabela de mensagens entrando na unidade FCM.	67
6.2 – Mensagens entrando na unidade FCM e suas prioridades.	73
6.3 – Tabela de mensagens e suas frequências relativas.....	78
6.4 – Número total de casas TDMA obtido a partir da Tabela 6.3.....	79
6.5 – Fatores de Multiplexação para as mensagens da Tabela 6.4.	80

1 INTRODUÇÃO

1.1 Contexto e Motivação do Trabalho

Sistemas computacionais que suportam missões em veículos aeroespaciais têm se especializado e aumentado em complexidade, tanto em hardware quanto em software. Por outro lado, a pressão por menor custo e menor risco tem mantido certo conservadorismo nas decisões de projeto, no sentido de que incertezas motivadas pela falta de um modelo melhor adaptado à situação real levam em geral a estimativas muito pessimistas que podem resultar em sobredimensionamento de componentes e aumento de custo. Contudo, é provável que esta condição não perdure por muito tempo. Novas tecnologias estão amadurecendo e ferramentas de simulação estão se tornando cada vez mais flexíveis e eficientes, permitindo antecipar problemas e encontrar soluções em ambientes virtuais, reduzindo custos e riscos de projeto.

Com a evolução de processadores e componentes eletrônicos, estes sistemas computacionais têm podido adotar uma arquitetura modular, simplificando seu projeto de hardware em alguns aspectos: as unidades ficam menores, têm aspectos físicos padronizados e compartilham componentes mecânicos e eletrônicos. Por outro lado, estas unidades em geral têm grande dependência de software para operar, fato gerador de grandes incertezas de projeto.

Em um sistema computacional embarcado, funções são distribuídas entre tarefas de software que precisam trocar dados com outras tarefas que não necessariamente executam na mesma unidade de hardware. Para que isto aconteça, as unidades precisam estar interligadas por um meio físico e lógico de comunicação de dados digitais eficiente. Ligações físicas entre unidades de um sistema embarcado podem ser ponto-a-ponto, mas esta topologia em geral leva a soluções pouco práticas em termos de peso, dimensões e percursos de cabos dentro do invólucro do sistema. Soluções onde o meio físico é compartilhado, como em sistemas modulares, em geral são mais práticas;

contudo, transferem a complexidade para entidades de software que ajudam a compor os subsistemas de comunicação de dados digitais. Este compartilhamento do meio físico de comunicação leva a um problema semelhante ao que foi resolvido matematicamente pela "Rate Monotonic Analysis" (RMA) (LIU;LAYLAND, 1973) que é: determinar rapidamente a condição suficiente, não necessária, para o agendamento ou escalonamento ("scheduling") de tarefas em um sistema computacional que possui uma única Unidade Central de Processamento (UCP).

De maneira análoga à RMA, tendo-se informação sobre parâmetros de execução das tarefas (máximo e mínimo tempo de execução, periodicidade se houver, tempo de resposta requerido) e sobre quais, quantos e com que frequência dados são trocados entre elas, imagina-se ser possível analisar a exeqüibilidade da transmissão e recepção de dados pelo subsistema de comunicação e determinar seus parâmetros fundamentais: tamanhos máximos de armazenadores intermediários ("buffers"), atrasos ("delays") e seus máximos e mínimos ("delay jitter").

Uma análise deste tipo permitirá tratar a mais importante característica na comunicação por rede em sistemas computacionais distribuídos embarcados em veículos aeroespaciais: o "determinismo" (a univocidade causal nas ocorrências de eventos), que contribuirá para a "previsibilidade" (a univocidade temporal nas ocorrências de eventos), entre a fase de projeto e a fase de execução, permitindo o correto dimensionamento do subsistema de comunicação de dados em harmonia com o subsistema de processamento de dados.

Situações assim permitem maior segurança na concepção do sistema de missão como um todo, reduzindo as incertezas e eliminando precauções que poderiam resultar em sobredimensionamento de componentes e, por consequência, em aumento de custo com aumento de peso e/ou volume.

No mundo aeronáutico, a adoção de arquiteturas modulares evoluiu para o que se chama de “Integrated Modular Avionics” (IMA) (RTCA, 2005). Sabe-se que métodos e ferramentas para assegurar o caráter determinístico de sistemas com arquitetura IMA e atender a critérios de certificação aeronáuticos foram desenvolvidos por algumas empresas especializadas, mas são pouco divulgados. Um conceito mais amplo que o conceito IMA é o que se chama “Integrated Modular Electronics” (IME), que considera soluções genéricas baseadas em integração de funções entre um mundo aviônico restrito e um “mundo exterior”, por exemplo, sistemas de entretenimento a bordo de aeronaves comerciais (VICTORIA PROJECT, 2001).

Há publicações que estudam arquiteturas distribuídas em sistemas de controle (SANZ;ÁRZÉN, 2003); nestes casos a preocupação é o desempenho do sistema de controle (avaliação da margem de fase, por exemplo) e não do canal de comunicação de dados entre o nó de controle e nós sensores ou atuadores. Sistemas de controle com arquitetura distribuída são considerados no escopo deste trabalho como geradores de requisitos funcionais, que orientariam o projeto de uma rede de comunicação de dados com características determinísticas adequadas ao desempenho esperado do sistema de controle como um todo.

1.2 Objetivo do Trabalho

O objetivo principal deste trabalho é exercitar algumas técnicas disponíveis para análise da transmissão e recepção de dados em uma rede de comunicação usando um modelo hipotético de um corretor de trajetória de vôo relativamente simples, de modo que algumas conclusões sobre parâmetros fundamentais da rede de comunicação envolvida possam ser obtidas sem grande esforço de cálculo. Como preparação para esta tarefa, serão apresentados alguns conceitos básicos fundamentais, os fundamentos da teoria do “Network Calculus” e noções de topologias e elementos de rede.

Convém lembrar que em projetos envolvendo processamento de dados em ambiente computacional, a determinação de tamanhos máximos de armazenadores intermediários (“buffers”), atrasos (“delay”) e suas variações (“delay jitter”) não têm solução analítica completa. Quando há alguma solução, ela em geral resolve um caso particular, restando estudar as consequências de se levar em conta outros fatores usando ferramentas complementares ou outro artifício conveniente. Neste trabalho, esta liberdade será amplamente exercitada.

Além do “determinismo” (a univocidade causal nas ocorrências de eventos), e da “previsibilidade” (a univocidade temporal nas ocorrências de eventos), outras duas características importantes, a “causalidade” (a ordenação temporal nas ocorrências de eventos do tipo causa e efeito) e o “sincronismo” (a simultaneidade nas ocorrências de eventos) (STANKOVIC, 1988), serão tratadas de maneira indireta como decorrência do caráter “determinístico” que estará sendo imprimido na comunicação de dados entre as unidades componentes do modelo do corretor de trajetória.

1.3 Organização do Trabalho

O Capítulo 2 apresenta conceitos básicos e faz uma revisão da literatura afim, em particular de fundamentos da teoria do “Network Calculus” necessária ao entendimento do trabalho e de topologias e elementos de rede. O Capítulo 3 faz uma avaliação do emprego da “Rate Monotonic Analysis” (RMA) na comunicação de dados por rede. O Capítulo 4 apresenta a formulação do problema e a descrição do modelo do “Corretor de Trajetória” usado como caso de estudo nas análises dos capítulos seguintes. O Capítulo 5 apresenta uma análise da comunicação de dados para o modelo do “Corretor de Trajetória” em uma implementação “federada”. O Capítulo 6 apresenta análises detalhadas em implementações “integradas” do “Corretor de Trajetória” usando recursos do “Network Calculus” e técnicas baseadas na multiplexação por divisão do

tempo e no modelo “Produtor-Consumidor”. O Capítulo 7 apresenta as conclusões e as sugestões para um possível prosseguimento deste trabalho.

2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA

2.1 Conceitos Básicos

As análises que serão feitas adiante pressupõem o conhecimento de alguns conceitos básicos que são reunidos a seguir:

- a) Sistema Computacional – A parte de um sistema maior que desempenha sua função através de um computador (MACHADO;MAIA, 2002).
- b) Sistema Computacional Embarcado – Um sistema computacional instalado dentro de um veículo e que desempenha funções essenciais à consecução da missão à qual o veículo se destina (“onboard computer system”) (ANSWERS, 2008).
- c) Rede de Comunicação – Tecnologia desenvolvida pelas empresas do ramo da computação para permitir troca de dados digitais entre dois ou mais computadores e seus dispositivos periféricos em alta velocidade (MACHADO;MAIA, 2002). É comum usar o termo “barramento” (“bus”) para identificar um canal de comunicação digital entre dois ou mais dispositivos eletrônicos capazes de processar dados digitalmente. Uma Rede de Comunicação pode ser formada por um ou mais “barramentos” (“buses”) compondo um sistema logicamente mais complexo, herdando total ou parcialmente as características dos “barramentos” que a compõem.
- d) Sistema Distribuído – Um sistema computacional que desempenha suas funções utilizando dois ou mais computadores e um ou mais canais de comunicação digital (MACHADO;MAIA, 2002).
- e) Determinismo – Fora do mundo da computação, “determinismo” significa “princípio ou teoria que condiciona os fenômenos naturais ou as decisões da vontade humana a certas condições anteriores” (LAROUSSE, 1992). No mundo da computação, este termo é usado

para associar univocidade ao comportamento de sistemas computacionais, isto é, fornecidas as mesmas condições e os mesmos estímulos iniciais, os eventos que se seguem são sempre os mesmos e acontecem sempre na mesma ordem, separados por intervalos de tempo que diferem entre si em ocorrências distintas por valores inferiores a um máximo especificado, em geral muito menor que o intervalo de amostragem (STANKOVIC, 1988).

- f) Causalidade – Fora do mundo da computação, “causalidade” significa “condição segundo a qual uma causa produz um efeito” (LAROUSSE, 1992). No mundo da computação este termo é usado para indicar a relação de causa-efeito entre eventos que se sucedem desde que um sistema é ligado. Eventos de software (execução de uma rotina ou tarefa) em geral se sucedem a eventos de hardware (interrupções causadas por dispositivos de entrada-saída ou pelo relógio interno do sistema) e/ou software (código privilegiado executado por rotinas do próprio sistema operacional) e suas interdependências, como por exemplo, o bloqueio de recursos comuns como memória e Unidade Central de Processamento (UCP) afetam a maneira como o sistema como um todo se comporta (BURNS;WELLINGS, 1997).
- g) Sincronismo – Também no mundo da computação, o termo “sincronismo” é usado para indicar a simultaneidade de ocorrências de dois eventos de hardware e/ou software (LAROUSSE, 1992) (BURNS;WELLINGS, 1997).
- h) Agendamento ou Escalonamento (“scheduling”) – Algoritmo executado dentro de um sistema computacional por uma entidade de software denominada “agendador” ou “escalador” (“scheduler”). Este algoritmo, através de um ou mais critérios, planeja dentre as tarefas ativas uma que será colocada para execução (MACHADO;MAIA, 2002) (BURNS;WELLINGS, 1997).

- i) “Rate Monotonic Analysis” (RMA) – Análise capaz de determinar rapidamente a condição suficiente, não necessária, para a realização do agendamento de tarefas periódicas em um sistema computacional com uma única CPU, admitindo que prioridades fixas são associadas as tarefas tanto mais alta quanto maior for sua frequência de ativação (LIU;LAYLAND, 1973).
- j) Atraso (“delay”) – Tempo medido entre ocorrências de dois eventos correlacionados em um sistema computacional (BURNS;WELLINGS, 1997).
- k) Variação de atraso (“delay jitter”) – Variação no tempo medido entre ocorrências de dois eventos correlacionados em um sistema computacional medido pela diferença entre seu maior e menor valor (BURNS;WELLINGS, 1997).
- l) Armazenador intermediário (“buffer”) – Dispositivo real ou virtual utilizado em sistemas computacionais para controlar um fluxo descontínuo de dados de entrada ou saída (MACHADO;MAIA, 2002) (BURNS; WELLINGS, 1997).

2.2 Fundamentos de “Network Calculus”

A teoria do “Network Calculus” foi introduzida por René Cruz (1991) e desenvolvida na última década (LE BOUDEC; THIRAN, 2004). Ela pode ser entendida como uma fundamentação matemática elaborada para explicar certos comportamentos em redes de comunicação de dados, em particular na Internet e em redes que usam tecnologia “Asynchronous Transfer Mode” (ATM) (IP/MPLS FORUM, 2008).

A abordagem de Cruz (1991) propõe limitar um fluxo de dados que entra em um sistema por uma função cumulativa. A Figura 2.1 mostra um fluxo de dados $R(t)$. Se um sistema é capaz de processar este fluxo de dados com a taxa $R1(t)$, a superposição das integrais de $R(t)$ e $R1(t)$ até t_1 permite antecipar o primeiro resultado do “Network Calculus”: a distância vertical entre as duas curvas é o tráfego retido (“backlog”) no sistema em qualquer instante t . Se $R(t)$ e $R1(t)$ são expressos em *bits/segundo*, então o tráfego retido será expresso em *bits*.

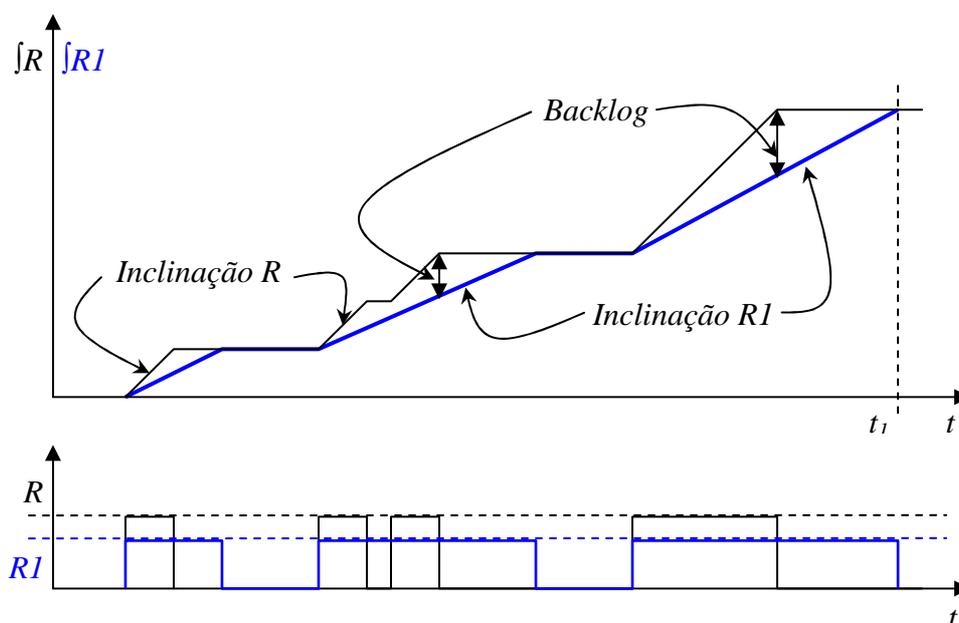


Figura 2.1 – Função cumulativa em “Network Calculus”.

Fonte: Adaptada de Cruz (1991).

Em particular, o “Network Calculus” define duas funções cumulativas importantes para análise de redes de comunicação: a Curva de Chegada (“Arrival Curve”) e a Curva de Serviço (“Service Curve”).

Diz-se que uma função α limita um fluxo de dados R entrando em um sistema se para todo $t \geq 0$ e $s \leq t$:

$$R(t) - R(s) \leq \alpha(t - s) \quad (2.1)$$

Satisfeita a relação, a função α é chamada Curva de Chegada para o fluxo R .

Considere um sistema que recebe um fluxo de dados R e produz um fluxo de saída R^* . Diz-se que este sistema oferece uma Curva de Serviço β se, para todo $t \geq 0$, existe algum $s \leq t$ tal que:

$$R^*(t) - R(s) \geq \beta(t - s) \quad (2.2)$$

Ambas funções são melhor ilustradas graficamente nas Figuras 2.2 e 2.3.

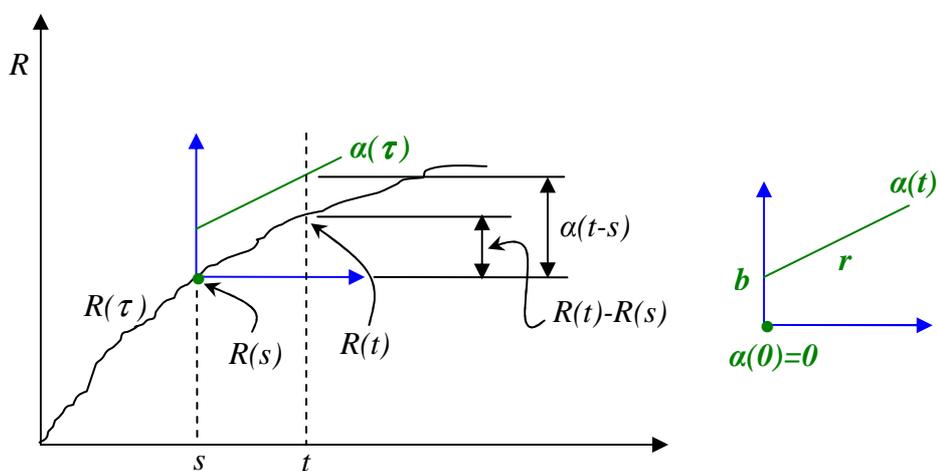


Figura 2.2 – Curva de Chegada em “Network Calculus”.

Usando o operador \otimes , pode-se reescrever as Equações 2.1 e 2.2 respectivamente nas seguintes formas, observando que as funções α e β são do tipo “wide-sense increasing”:

$$R(t) \leq \inf \{ \alpha(t-s) + R(s) \} = (\alpha \otimes R)(t), \quad 0 \leq s \leq t \quad (2.4)$$

$$R^*(t) \geq \inf \{ \beta(t-s) + R(s) \} = (\beta \otimes R)(t), \quad 0 \leq s \leq t \quad (2.5)$$

Uma parte interessante da teoria do “Network Calculus” materializa-se em três importantes resultados: o limite da quantidade de tráfego retido no sistema (“backlog bound”), o limite do atraso (“delay bound”) e o limite do fluxo de saída (“output flow”).

O limite superior da quantidade de tráfego retido em um sistema que oferece uma Curva de Serviço β a um fluxo de entrada R limitado por uma Curva de Chegada α e produz um fluxo de saída R^* é definido por:

$$R(t) - R^*(t) \leq \sup \{ \alpha(s) - \beta(s) \}, \quad s \geq 0 \quad (2.6)$$

Este limite é o maior desvio vertical medido entre as funções α e β como ilustrado na Figura 2.4.

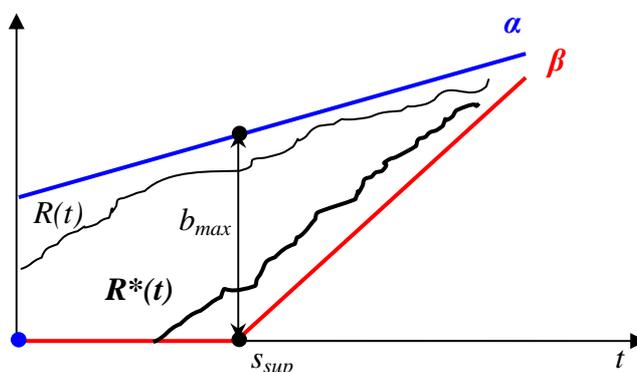


Figura 2.4 – Limite superior do tráfego retido em um sistema.

O atraso $d(t)$ percebido por um fluxo de entrada limitado por uma Curva de Chegada α que atravessa um sistema que oferece uma Curva de Serviço β é definido por:

$$d(t) = \inf [\tau \geq 0 \mid \alpha(t) \leq \beta(t + \tau)] \quad (2.7)$$

O maior atraso é o maior desvio horizontal entre as duas funções α e β , e é definido por:

$$d_{max} = \sup [d(t)], t \geq 0 \quad (2.8)$$

Este limite superior pode ser melhor visualizado graficamente na Figura 2.5.

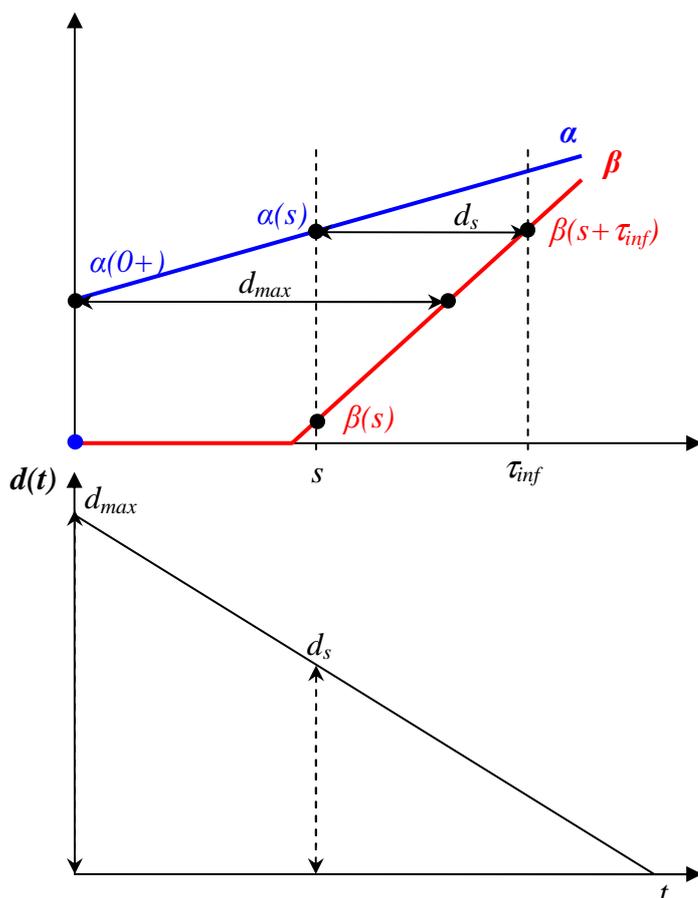


Figura 2.5 – Atraso máximo em um fluxo de entrada.

Para obtenção do limite do fluxo de saída, é necessário definir primeiro a operação de *deconvolução*.

Se duas funções f e g são “wide-sense increasing”, a *deconvolução* em álgebra “min-plus” é definida por:

$$(f \ominus g) (t) = \sup \{ f(t + u) - g(u) \}, u \geq 0 \quad (2.9)$$

Se f e g são infinitas para algum t , então a operação de *deconvolução* não é definida. Ao contrário da *convolução*, a *deconvolução* em $t < 0$ não é necessariamente 0.

O fluxo de saída de um sistema que oferece uma Curva de Serviço β a um fluxo de entrada limitado por uma Curva de Chegada α é limitado por uma outra Curva de Chegada α^* definida por:

$$\alpha^* = (\alpha \odot \beta) \quad (2.10)$$

Para ilustrar melhor as operações de *convolução* e *deconvolução*, é conveniente introduzir duas funções do tipo “wide-sense increasing” que serão usadas no contexto deste trabalho.

A função “Affine” $\alpha_{r,b}$ assume valores:

$$\alpha_{r,b}(t) = 0, t = 0 \quad (2.11)$$

$$\alpha_{r,b}(t) = r \cdot t + b, t > 0, b > 0$$

A função “Rate-Latency” $\beta_{R,T}$ assume valores:

$$\beta_{R,T}(t) = 0, 0 \leq t \leq T \quad (2.12)$$

$$\beta_{R,T}(t) = R[t - T]^+$$

Ambas estão representadas graficamente na Figura 2.6.



Figura 2.6 – Função “Affine” (à esquerda) e função “Rate-Latency” (à direita).

Nas Figuras 2.7 e 2.8 estão representadas a *convolução* e a *deconvolução* de funções “Affine” $\alpha_{r,b}$ e “Rate-Latency” $\beta_{R,T}$ ($R > r$).

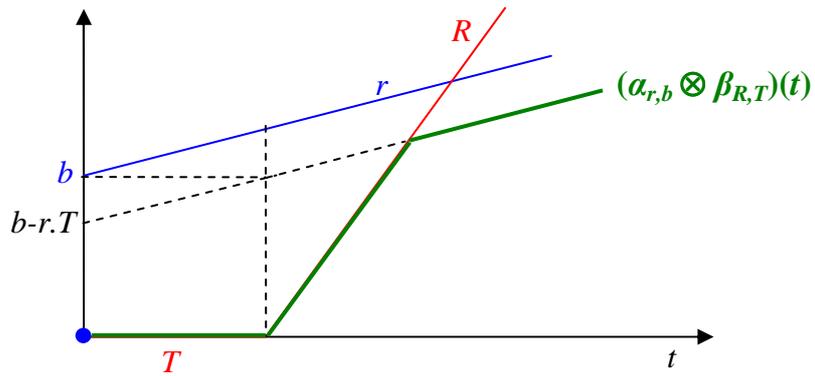


Figura 2.7 – Convolução de funções “Affine” $\alpha_{r,b}$ e “Rate-Latency” $\beta_{R,T}$.

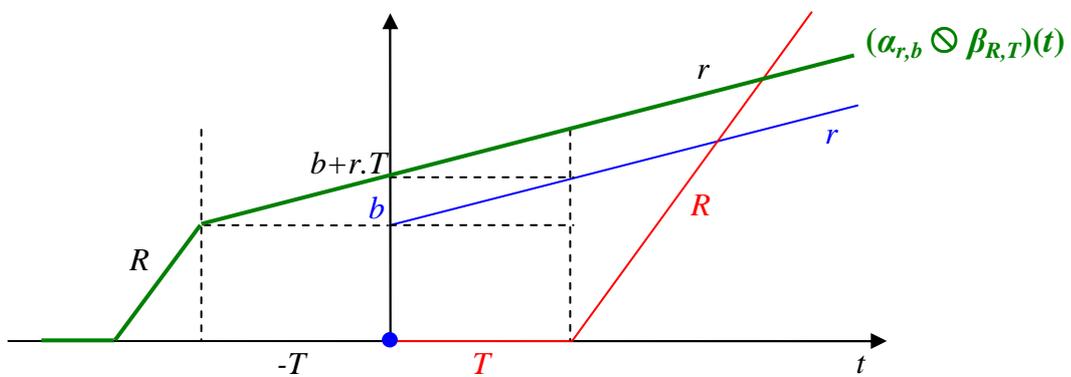


Figura 2.8 – Deconvolução de funções “Affine” $\alpha_{r,b}$ e “Rate-Latency” $\beta_{R,T}$.

É conveniente também mostrar os três limites (“backlog bound”, “delay bound” e “output flow”) obtidos para um sistema cujo fluxo de entrada é limitado por uma Curva de Chegada do tipo “Affine” $\alpha_{r,b}$ e cujo fluxo de saída é limitado por uma Curva de Serviço do tipo “Rate-Latency” $\beta_{R,T}$ ($R > r$).

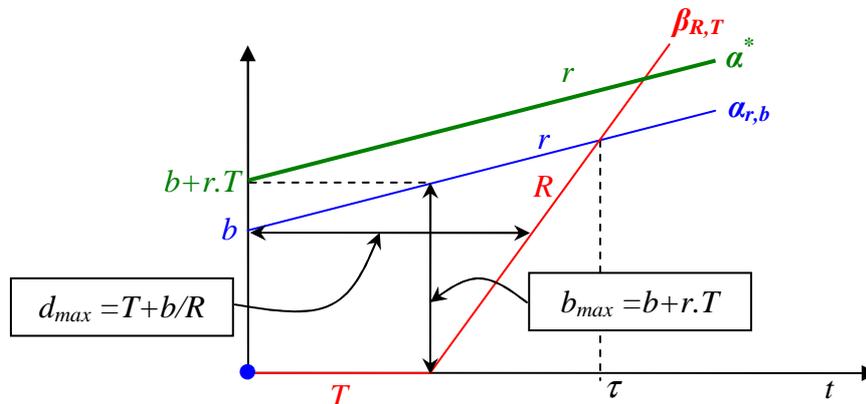


Figura 2.9 – Os três limites em funções “Affine” $\alpha_{r,t}$ e “Rate-Latency” $\beta_{R,T}$.
 Fonte: Adaptada de Le Boudec, Thiran (2994).

Na Figura 2.9, o limite superior do tráfego b_{max} retido no sistema (“backlog bound”) é igual a $b+r.T$, o limite superior do atraso d_{max} (“delay bound”) é igual a $T+b/R$, e o fluxo de saída é limitado pela Curva de Chegada α^* cuja equação da reta é $b+(t+T)r$. Notar que a validade das equações para os dois primeiros limites se limita à região $0 \leq t \leq \tau$, τ satisfazendo a relação $\beta_{r,t}(\tau) = \alpha_{r,b}(\tau)$ se $R > r$.

Maiores detalhes sobre o “Network Calculus” podem ser encontrados principalmente em Le Boudec, Thiran (2004). Um estudo prático da aplicação desta teoria feito de uma maneira bem didática pode ser encontrado em Morais (2005).

Como complementação, será apresentado em seguida um estudo sobre a determinação do fluxo de saída usando “rate functions”, como descrito por René Cruz (1991) (ver Figura 2.1) para um sistema que oferece uma Curva de Serviço do tipo “Rate-Latency” $\beta_{R,T}$ a um fluxo de entrada limitado por uma Curva de Chegada do tipo “Affine” $\alpha_{r,b}$ ($R > r$).

As análises feitas com a integral de um fluxo de dados no tempo trazem o benefício de “suavizar” o efeito instantâneo de “rajadas” (“bursts”) de dados, que de outra maneira pareceriam grandes perturbações em um gráfico que

descreve variações do fluxo no tempo. Isto é especialmente vantajoso quando se faz análise de desempenho, por exemplo, em um serviço público Internet, onde a frequência na qual chegam requisições é totalmente aleatória e apresenta picos e vales muito distintos (MORAIS, 2005). Contudo, em situações mais triviais como a que será apresentada, uma análise partindo de “rate functions” traz o benefício de aumentar a compreensão do problema e de como o “Network Calculus” encara os três limites (“backlog bound”, “delay bound” e “output flow”).

Na Figura 2.10a, a Curva de Chegada “Affine” $\alpha_{r,b}$ está representada por uma “rate function” de taxa 0 em $t = 0$, um “impulso instantâneo” (representado uma “rajada” ou um “burst”) de tamanho b em $t = 0+$, e uma constante r para $t > 0$. Na Figura 2.10b, a Curva de Serviço “Rate-Latency” $\beta_{R,T}$ está representada por uma “rate function” de taxa 0 entre $0 \leq t \leq T$ e uma constante R para $t > T$.

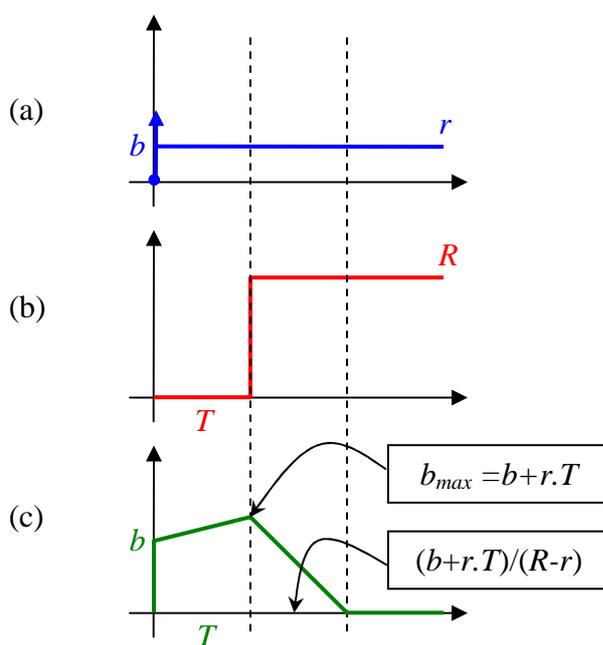


Figura 2.10 – Evolução do tráfego retido usando “rate functions”.

A Figura 2.10c mostra a evolução no tempo do tráfego retido no sistema, que atinge um máximo igual a $b+r.T$ em $t = T$, e cai a zero em $t = T + (b+r.T)/(R-r)$. Nas expressões anteriores, o termo $r.T$ representa o tráfego adicional retido no

sistema até $t = T$, além da “rajada” inicial de valor b , já que a Curva de Serviço serve o fluxo de entrada com taxa 0 até $t = T$. Em $t > T$, o tráfego retido no sistema cai do seu máximo até 0 com uma taxa $(R - r)$.

Para $t > T + (b+r.T)/(R-r)$, não há mais tráfego retido no sistema, dados entram com uma taxa r e podem ser servidos com uma taxa R suposta maior que r . Como não é possível retirar dados do sistema a uma taxa maior do que dados entram no sistema, admite-se que a partir deste ponto o fluxo de saída é limitado pelo fluxo de entrada.

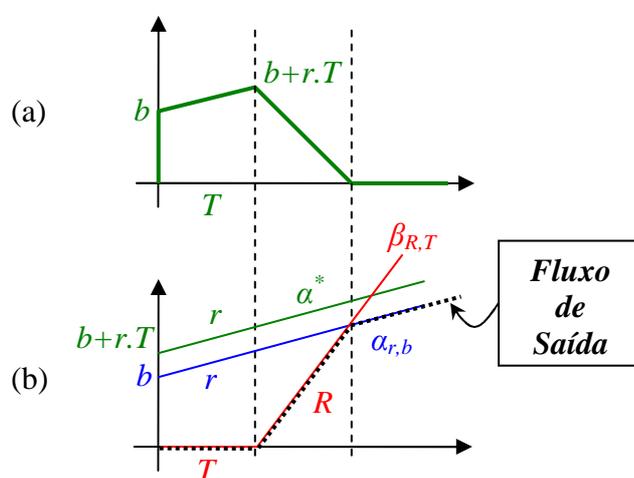


Figura 2.11 – Fluxo de Saída usando “rate functions”.

Aparentemente, o limite do fluxo de saída obtido por deconvolução da Curva de Chegada e da Curva de Serviço no “Network Calculus” é um resultado mais conservador, porque, na Figura 2.11b, o fluxo de saída obtido por “rate functions” (linha pontilhada) está sempre abaixo da Curva de Chegada α^* , terceiro limite do “Network Calculus” para as Curvas de Chegada e Serviço “Affine” e “Rate-Latency” em questão. A Figura 2.11a é a mesma Figura 2.10c, repetida apenas por uma questão de clareza.

A Figura 2.12 ilustra uma situação onde a Curva de Chegada é $\alpha(t) = r.t$ e a Curva de Serviço é $\beta(t) = R.t$, ($R > r$) em um sistema observado até $t = T$. O

resultado da deconvolução $\alpha^* = \alpha \otimes \beta$ coincide com α . O mesmo se verifica se na Figura 2.9 faz-se $T = 0$.

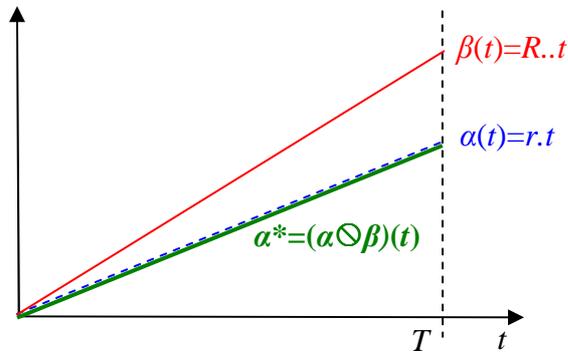


Figura 2.12 – Limite do fluxo de saída para um caso $R > r$.

Fonte: Adaptada de Morais (2005).

Em uma situação semelhante fazendo $R < r$, a teoria produz um resultado interessante representado na Figura 2.13.

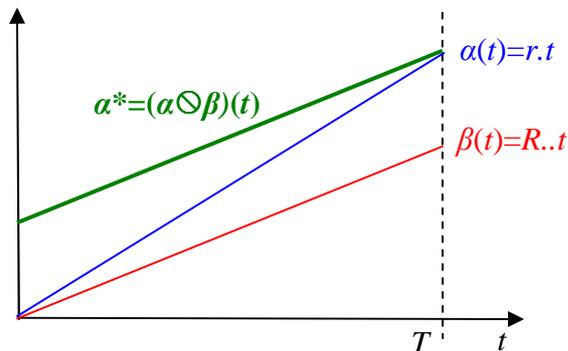


Figura 2.13 – Limite do fluxo de saída para um caso $R < r$.

Fonte: Adaptada de Morais (2005).

Note-se que o resultado da deconvolução α^* admite uma “rajada” em $t = 0$ igual ao tráfego acumulado até $t = T$ (lembrar que $R < r$) e realmente limita o fluxo de saída R até este instante.

2.3 Revisão de Topologias e Elementos de Rede

Os próximos capítulos deste trabalho dependem do entendimento de algumas implementações de redes de comunicação e do funcionamento de um de seus componentes-chave.

Quando duas ou mais unidades capazes de processamento digital são interligadas em rede, elas compartilham um único meio físico de comunicação, seja ele um cabo ou um outro dispositivo de hardware. A maneira como estas unidades se ligam a este meio físico é costumeiramente chamada no ramo da Informática de “topologia”, termo herdado do ramo da Matemática que estuda propriedades qualitativas dos espaços (H. Poincaré, estudos entre 1895 e 1904) (ANSWERS, 2008).

A literatura especializada (MARTIN, 1989) cita três topologias básicas que sozinhas ou combinadas cobrem a maioria dos casos comerciais: “barramento” (“bus”), “anel” (“ring”) e “estrela” (“star”).

Na topologia em “barramento”, todas as unidades estão ligadas diretamente a um canal de comunicação comum, em geral um cabo metálico blindado aberto com terminações resistivas. Na topologia “anel”, as unidades também estão ligadas diretamente a um canal de comunicação comum, mas neste caso o meio físico forma um anel fechado (“closed ring” ou “loop”). Na topologia “estrela”, as unidades estão ligadas a um controlador central que é responsável por intermediar toda a comunicação, ou seja, cabe a ele receber mensagens de cada unidade e retransmití-las para que cheguem a uma ou mais unidades de destino. A Figura 2.14 ilustra as três topologias.

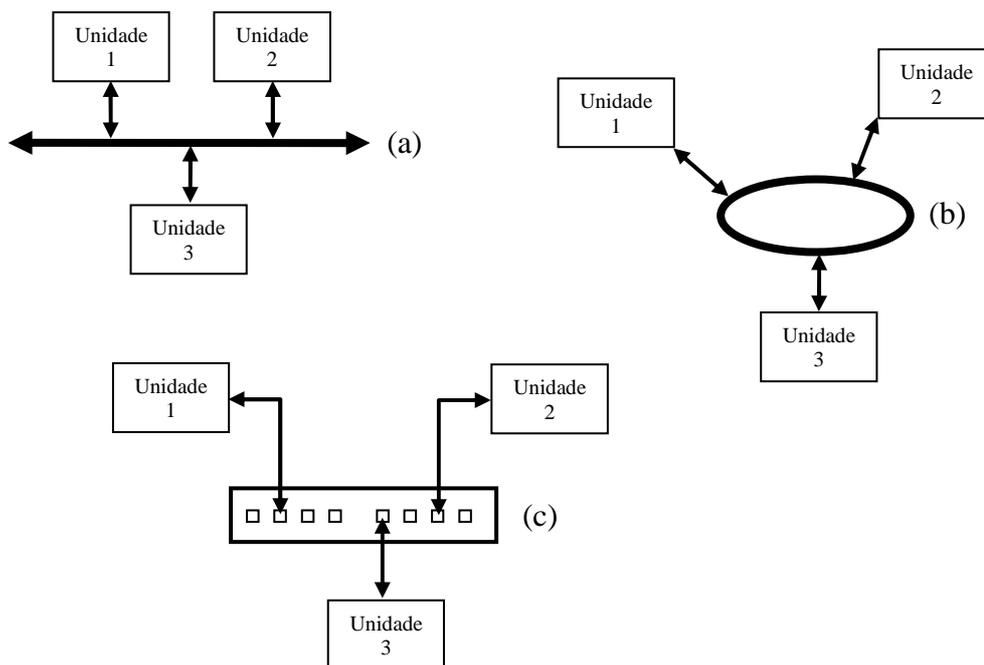


Figura 2.14 – Topologias de Rede: (a) “barramento”, (b) “anel”, (c) “estrela”.

O controlador central presente na topologia “estrela” é de particular interesse para este trabalho.

Entre 1979 e 1982, a Xerox, a Digital Equipment Corporation (DEC) e a Intel se juntaram para padronizar, promover e comercializar produtos derivados de uma tecnologia desenvolvida pela Xerox em 1973 para interligar estações de trabalho através de uma rede de comunicação de dados digitais (SEIFERT, 2000). Esta tecnologia denominada “Ethernet” foi base para construção da norma IEEE 802.3 (IEEE, 2005) e colaborou para a popularização do que é conhecido hoje como Internet.

Originalmente, a Ethernet interligava computadores em uma topologia “barramento” usando como meio físico um cabo coaxial terminado em ambas as pontas (“yellow cable”). A partir de 1984, dispositivos foram introduzidos para estender uma rede Ethernet além do limite do meio físico original, tanto em comprimento (inicialmente 500 metros para uma velocidade de 10

Megabits/segundo) quanto em número de pontos de conexão (199). Um destes dispositivos denominado “bridge” permitia interligar e filtrar o tráfego entre dois segmentos de rede, aliviando o problema de congestionamento e a conseqüente queda de desempenho quando o número de transmissões se aproximava da capacidade do meio físico. Na década de 1990, a tecnologia de semi-condutores evoluiu a ponto de permitir a construção de “bridges” capazes de interligar muitos segmentos de rede sem introduzir grandes latências de transmissão. Estas novas “bridges” foram batizadas de “switches” e continuaram a evoluir incorporando novas funcionalidades até se tornarem um elemento essencial na moderna infraestrutura de rede Ethernet. Convém lembrar que “switching” não é privilégio de redes Ethernet: redes ATM foram concebidas usando os conceitos de “cell switching” e de “circuit switching” (IP/MPLS FORUM, 2008).

Com a introdução de “switches” em redes Ethernet, a topologia que antes era em “barramento” passou a ser em “estrela”, com o “switch” fazendo o papel de controlador central. Inicialmente, “switches” operavam em modo “half-duplex”, onde transmissão e recepção são operações mutuamente exclusivas; mas com a introdução da tecnologia “full-duplex” em redes Ethernet, onde transmissão e recepção podem acontecer simultaneamente, em pouco tempo todos os “switches” do mercado operavam “full-duplex” com enormes vantagens de desempenho.

“Switches” Ethernet modernos como os da Figura 2.15 (ENTERASYS, 2008) possuem poder de processamento comparável ao de um servidor de classe corporativa do final da década de 1990 em apenas um módulo. Contudo, o projeto básico é o mesmo desde que a Kalpana Corporation e a Grand Junction Networks lançaram os primeiros exemplares no início da década de 1990 (SEIFERT, 2000).



Figura 2.15 – “Switches” Matrix® N-Series da Enterasys Networks, Inc.

A Figura 2.16 ilustra os três elementos lógicos essenciais de um “switch” Ethernet: as filas de entrada (“input queues”), o dispositivo chaveador (“switching fabric”) e as filas de saída (“output queues”). Cada fila de entrada ou saída deve atender a uma porta física do “switch”, enquanto que o dispositivo chaveador se encarrega de mover mensagens da fila de entrada para uma ou mais filas de saída. Notar que a operação “half-duplex” ou “full-duplex” de um “switch” afeta diretamente a maneira como filas de entrada e saída são processadas, com impacto no seu desempenho e, por conseguinte, nas latências observadas na transmissão de mensagens.

Normalmente, é desejável que mensagens fiquem na fila de entrada por apenas um tempo ínfimo e que elas cheguem o mais rapidamente possível à fila de saída, minimizando sua “estadia” dentro do “switch”. Como mensagens recebidas em uma porta podem ter como destino uma ou mais portas, não é difícil perceber que as filas de saída podem ser um ponto de congestionamento em “switches” Ethernet. Na verdade, filas de saída são pontos de congestionamento em “switches” com tecnologia “shared memory” e filas de entrada são pontos de congestionamento em “switches” com tecnologia “crossbar” (ou “switching matrix”) (SEIFERT, 2000).

Para lidar com situações de acúmulo de tráfego na saída, foram criados alguns artifícios que permitem ao operador da rede controlar melhor o desempenho de “switches” em uma rede Ethernet de maior porte.

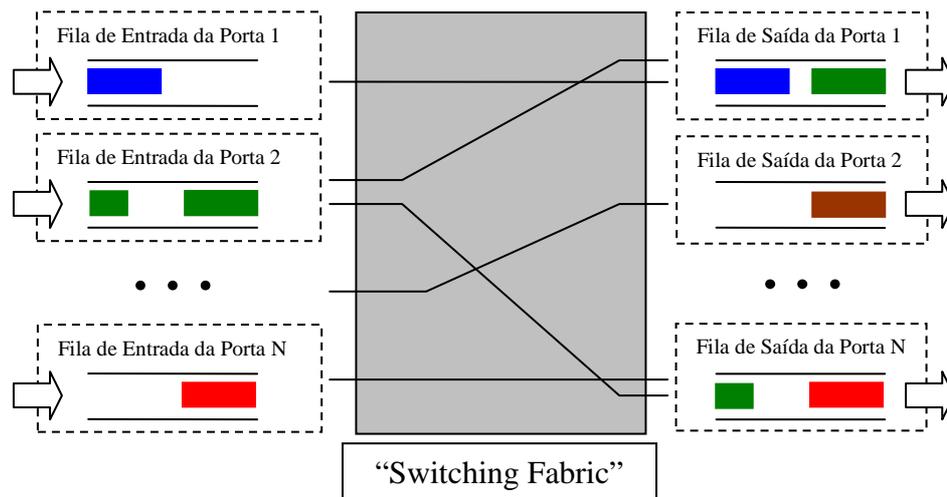


Figura 2.16 – Elementos lógicos de um “switch” Ethernet.

Um destes artifícios se assemelha muito àquele usado para agendar tarefas de forma ordenada em um sistema computacional de uma única Unidade Central de Processamento (UCP): distribuir as mensagens em sub-filas de prioridades diferentes e proceder à remoção de mensagens das sub-filas começando pela de prioridade mais alta.

Georges et al (2005) procuraram modelar atrasos em “switches” Ethernet usando elementos de rede definidos por Cruz (1991). Um outro trabalho (ALENA et al, 2006) caracterizou em laboratório o desempenho de um “switch” Ethernet comercial com resultados muito interessantes, por exemplo, que a latência medida na passagem de uma mensagem através de um “switch” do tipo “shared memory” escala com o tamanho da mensagem de forma não linear.

Mais adiante neste trabalho, o processamento da fila de saída de um “switch” será modelado usando o conceito de Curva de Serviço do “Network Calculus”, de maneira semelhante ao que foi feito por Grieu (2004).

3 AVALIAÇÃO DO USO DA ABORDAGEM “RATE MONOTONIC” NA COMUNICAÇÃO DE DADOS POR REDE

O resultado surpreendentemente imediato do teste de agendabilidade proposto por Liu e Layland (1973) parece atraente para rapidamente resolver o problema de verificar a ocupação de um canal de comunicação de dados compartilhado. Esta verificação, feita antecipadamente em fase de projeto, poderia assegurar o caráter determinístico da comunicação de dados entre unidades de um sistema embarcado.

No trabalho de Liu e Layland, o recurso compartilhado é uma única Unidade Central de Processamento (“Central Processing Unit”, CPU), que é disputada por tarefas com diferentes prioridades, tempos de execução e períodos (ou frequências) de ativação de modo a cumprir suas metas de agendamento, ou seja, completar sua execução antes que ela seja ativada novamente. Nesta situação, um “agendador” interrompe a execução da tarefa corrente e ativa uma outra de maior prioridade fazendo o que se chama de “agendamento preemptivo”. O trabalho de Liu e Layland pressupõe um agendamento preemptivo em que tarefas periódicas são agendadas com prioridades fixas maiores tanto quanto menores forem seus períodos de ativação. A este particular critério de associar prioridades a tarefas chama-se “Rate Monotonic”.

A “Utilização” (U), fração de tempo em que uma tarefa ocupa a CPU, é definida como o quociente entre o “Tempo de Execução” (C), tempo atribuído a uma execução completa de uma tarefa, e o “Período” (T), inverso da frequência de ativação de uma tarefa. Liu e Layland mostraram que um teste de agendabilidade pode ser obtido considerando apenas a utilização combinada do conjunto de tarefas observado.

Se a seguinte condição é satisfeita, então todas as N tarefas completarão sua execução antes que elas sejam novamente ativadas:

$$\sum_{i=1}^N \left(\frac{C_i}{T_i} \right) < N(2^{1/N} - 1) \quad (3.1)$$

Observe-se que a expressão à esquerda na Equação 3.1 é a utilização combinada (soma dos quocientes C/T para tarefas de 1 a N) e que a expressão à direita depende apenas do número N de tarefas observadas. Quando este número tende a infinito, o limite desta expressão tende assintoticamente para o valor 0,693147 ($\ln 2$). Assim, qualquer conjunto de tarefas cuja utilização combinada é inferior a 69,3% é agendável por um esquema de agendamento preemptivo cujas prioridades sejam fixas e associadas a tarefas pelo critério “Rate Monotonic”, resguardadas as seguintes condições de contorno: i) as tarefas têm tempos de execução e períodos de ativação fixos; ii) as tarefas devem completar sua execução dentro um intervalo de tempo no máximo igual ao seu período de ativação; iii) as tarefas são independentes umas das outras, ou seja, não compartilham qualquer recurso e não precisam executar segundo uma ordem de precedência; iv) a utilização combinada do conjunto de tarefas observado é menor ou igual a 1 (100%).

A Tabela 3.1 ilustra um conjunto de tarefas que passa no teste de agendabilidade de Liu e Layland.

Tabela 3.1 – Tabela de tarefas para teste de agendabilidade.

Tarefa	Período T	Tempo de Execução C	Prioridade P	Utilização U
1	60	20	1	0,333
2	40	10	2	0,250
3	30	5	3	0,166

A utilização combinada do conjunto de 3 tarefas da Tabela 3.1 resulta em 0,750, inferior a 0,779 resultado de $3(2^{1/3} - 1)$, portanto passa no teste de agendabilidade se as prioridades forem associadas segundo o critério “Rate Monotonic” (maior prioridade 3 associada à tarefa de menor período 30).

Este teste de agendabilidade deve ser entendido como condição suficiente, não necessária, para que o conjunto de tarefas observado seja efetivamente agendável usando o critério “Rate Monotonic”. Portanto, existem outros testes como, por exemplo, a Análise do Tempo de Resposta (“Response Time Analysis”) (BURNS;WELLINGS, 1997) que permitem um resultado mais preciso.

Muitas vezes é conveniente mostrar o agendamento de tarefas na linha do tempo em formato de gráficos de Gantt, como na Figura 3.1. Pode-se mostrar que, se as tarefas do conjunto observado são ativadas no mesmo instante (não significa que elas executam no mesmo instante), é suficiente observar o conjunto por um período igual ao da tarefa de maior período (BURNS;WELLINGS, 1997).

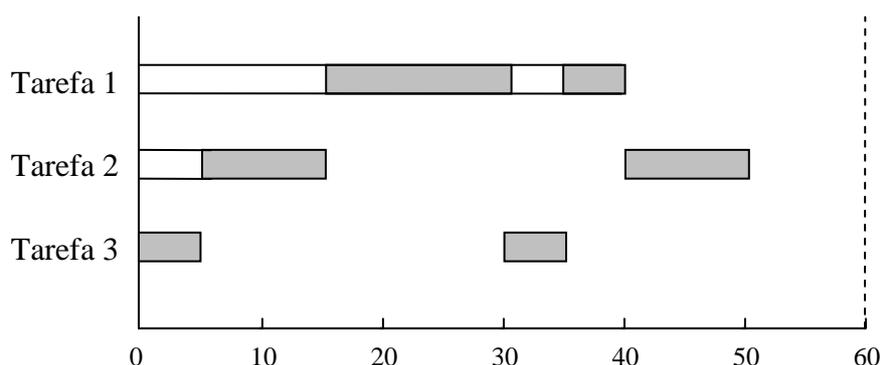


Figura 3.1 – Gráfico de Gantt para o conjunto de tarefas da Tabela 5.1.

Liu e Layland (1973) definem dois conceitos importantes para a interpretação correta da situação apresentada na Figura 3.1: i) o “tempo de resposta” de uma tarefa é o tempo transcorrido entre sua ativação e o término de sua execução;

ii) o “instante crítico” de uma tarefa é o instante de ativação desta tarefa que produz o maior tempo de resposta. Além disto, Liu e Layland mostram que o “instante crítico” de qualquer tarefa ocorre quando ela é ativada simultaneamente com todas as tarefas de maior prioridade que a própria. Na Figura 3.1, o “instante crítico” da Tarefa 1 ocorre em $t = 0$ quando ela é ativada simultaneamente com as Tarefas 2 e 3 de maior prioridade, e o “instante crítico” da Tarefa 2 também ocorre no instante $t = 0$ quando ela é ativada simultaneamente com a Tarefa 3 de maior prioridade. Assim, o “instante crítico” do conjunto de tarefas ocorre quando todas as tarefas são ativadas simultaneamente, como na Figura 3.1.

Lembrando que as tarefas disputam uma única CPU, observa-se que a Tarefa 2 só inicia execução no instante 5 por ter sido impedida pela Tarefa 3 que tem maior prioridade. A Tarefa 1 só inicia execução no instante 15 por ter sido impedida pelas Tarefas 2 e 3 que têm maior prioridade; depois ela é interrompida novamente pela Tarefa 3 entre os instantes 30 e 35; e ela só completa a execução no instante 40, felizmente antes do instante 60 da sua próxima ativação.

Uma situação onde 3 unidades compartilham um mesmo canal de comunicação como ilustrado na Figura 3.2 para transmissão de mensagens com periodicidade fixa de certa forma se assemelha à situação descrita anteriormente para o conjunto de tarefas da Tabela 3.1. Se o canal de comunicação é compartilhado, o meio físico é único e, portanto, é necessário estabelecer algum protocolo para que as transmissões ocorram de maneira ordenada.

Um protocolo simples de acesso ao meio físico do canal de comunicação poderia ser: i) unidades enviam mensagens com prioridades diferentes auto-identificadas no próprio corpo da mensagem; ii) se uma unidade identifica a transmissão de uma mensagem de maior prioridade em andamento ao acessar

o canal de comunicação, ela interrompe sua própria transmissão imediatamente.

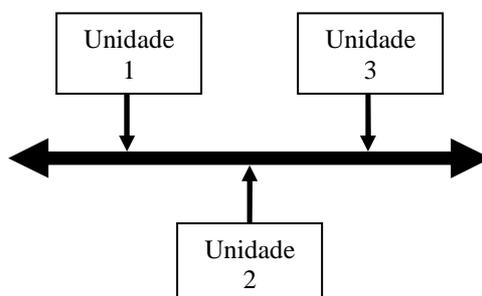


Figura 3.2 – Três unidades compartilhando um canal de comunicação.

Este protocolo, uma simplificação daquele especificado em (CAN, 1991), se assemelha ao “agendamento preemptivo” mencionado anteriormente a menos da ausência da figura do agendador. Por conseguinte, parece tentador usar de uma forma adaptada o mesmo teste de agendabilidade de Liu e Layland e presumir que, se ele for satisfeito, todas as mensagens poderiam ser transmitidas com sucesso.

Esta adaptação se materializaria pela substituição do “Tempo de Execução” (C) pelo “Tempo de Transmissão” (Tx), equivalente ao quociente entre o comprimento da mensagem (possivelmente em *bits*) e a velocidade com que o meio físico opera (possivelmente em *bits/segundo*). Feito isto, a Tabela 3.1 se transformaria na Tabela 3.2 repetindo os resultados de “Utilização” (U) sem alteração.

Tabela 3.2 – Tabela de mensagens para teste de transmissibilidade.

Mensagem	Período T	Tempo de Transmissão Tx	Prioridade P	Utilização U
1	60	20	1	0,333
2	40	10	2	0,250
3	30	5	3	0,166

Como a utilização combinada do canal de comunicação pelas mensagens da Tabela 3.2 é o mesmo das tarefas da Tabela 3.1, pode-se ficar tentado a dizer que todas as 3 mensagens serão transmitidas com sucesso se o critério de atribuição de prioridades de transmissão for “Rate Monotonic”.

Contudo, há uma diferença crucial entre agendar tarefas em um sistema com uma única CPU e transmitir mensagens em um canal de comunicação compartilhado: ao contrário de tarefas, transmissões uma vez efetivamente iniciadas não podem ser interrompidas. Em um meio físico que permite arbitramento nas transmissões de mensagens (CAN, 1991), mensagens de menor prioridade interrompem sua transmissão somente se identificarem a transmissão de uma outra de maior prioridade dentro de um período mínimo representado por um “preâmbulo” transmitido logo no início da mensagem. Uma vez passado este período, o meio físico é percebido como “ocupado” e outras transmissões ficam suspensas até o final da transmissão da mensagem corrente, representado em geral por um período mínimo de “silêncio” (ausência de sinal) no meio físico. Qualquer interrupção na transmissão entre o “preâmbulo” e o final da mensagem em geral é considerado um erro e o fragmento de mensagem é descartado.

Desta maneira, o gráfico de Gantt da transmissão das mensagens da Tabela 3.2 é diferente da Figura 3.1.

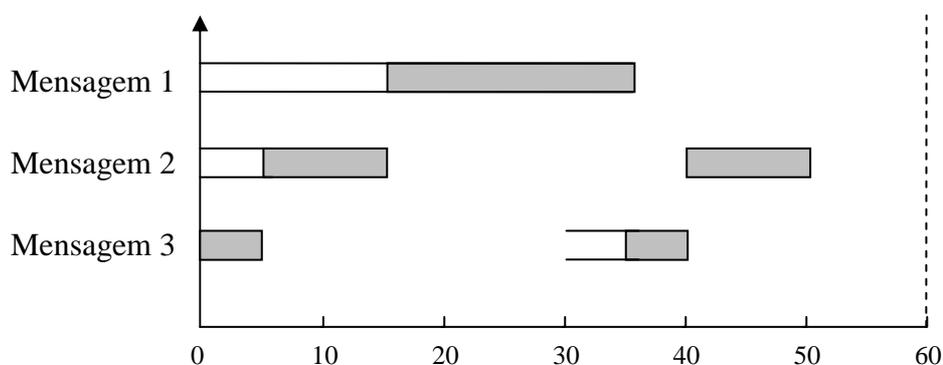


Figura 3.3 – Gráfico de Gantt para o conjunto de mensagens da Tabela 3.2.

Na Figura 3.3, pode-se ver claramente que a Mensagem 3, que tem período 30, só conseguiu ser transmitida no instante 35 porque a transmissão da Mensagem 1 de menor prioridade ainda estava em andamento. Esta situação (a rigor uma inversão de prioridades) viola claramente uma das condições que precisam ser respeitadas para que o teste de agendabilidade de Liu e Layland possa ser aplicado, mais precisamente aquela que pressupõe “agendamento preemptivo”, no sentido que tarefas de maior prioridade interrompem a execução de tarefas de menor prioridade e não o contrário. Esta violação também afeta o uso imediato dos conceitos de “tempo de resposta” e “instante crítico” mencionados acima.

Neste caso de transmissão periódica de mensagens por um canal de comunicação compartilhado, parece que é suficiente observar a seguinte regra para que todas as mensagens sejam transmitidas com a frequência programada: *a soma total dos tempos de todas as transmissões não pode ser maior que o menor período*. No caso das mensagens da Tabela 3.2, a soma total das transmissões $(20 + 10 + 5) = 35$ é evidentemente maior que o menor período 30. Observar que esta regra simples em princípio vale independentemente da associação de prioridades às transmissões de mensagens.

Mais à frente neste trabalho será apresentado um protocolo de acesso ao meio físico de transmissão que permite que mensagens periódicas sejam transmitidas sempre com as frequências programadas.

Por enquanto, pode-se concluir que o teste de agendabilidade de Liu e Layland não pode ser imediatamente adaptado a transmissões periódicas de mensagens por um canal de comunicação compartilhado, pelo fato de que transmissões de mensagens, ao contrário de execução de tarefas, uma vez efetivamente iniciadas, não podem ser interrompidas.

Como consequência, adaptações de outras análises como o “Tempo de Resposta” (“Response Time Analysis”) (BURNS;WELLINGS, 1997) mesmo em sistemas distribuídos (“End-to-End Response Time”) (SUN;LIU, 1996) (LIU, 2000) também ficam prejudicadas, porque todas pressupõem agendamento preemptivo na execução de tarefas.

4 FORMULAÇÃO DO PROBLEMA E DESCRIÇÃO DO MODELO DO CORRETOR DE TRAJETÓRIA

4.1 Formulação do Problema

O objetivo principal deste trabalho é a Análise do Determinismo na Comunicação por Rede em Sistemas Computacionais Embarcados em Veículos Aeroespaciais.

Dentre as opções disponíveis para realização deste objetivo, o uso de simulação usando modelos disponíveis de redes de comunicação como CAN ou Ethernet precisaria passar por uma validação destes modelos em um cenário conhecido, de modo a criar confiança nos resultados dos quais dependeria todo o presente estudo. O uso de experimentação dependeria da disponibilidade de equipamentos como um “switch” Ethernet e de ferramentas de análise de tráfego de dados neste tipo de rede. Neste caso, levantar as características funcionais do equipamento usado (no caso de um “switch” Ethernet, o tempo de comutação de pacotes em função do tamanho, por exemplo) seria fundamental para que fossem eliminadas das análises quaisquer perturbações indesejáveis causadas por alguma delas.

A estratégia escolhida para este trabalho foi a de exercitar algumas técnicas disponíveis na literatura e validadas em outros cenários reais para análise da transmissão e recepção de dados em uma rede de comunicação usando um modelo hipotético, mas representativo, de modo que algumas conclusões sobre os parâmetros fundamentais desta rede de comunicação pudessem ser obtidas sem grande esforço de cálculo.

O modelo escolhido foi o de um corretor de trajetória relativamente simples, que é descrito a seguir. Ele tem os componentes de um sistema com esta funcionalidade, e o volume e frequência com que as transmissões de dados ocorrem entre eles se aproximam muito de um sistema real.

4.2 Descrição do Modelo do “Corretor de Trajetória”

Para este trabalho, imaginou-se um modelo simples de um corretor de trajetória implementado em um sistema distribuído embarcado em um veículo não-tripulado que viaja pela atmosfera.

As unidades componentes deste modelo são:

- a) Unidade de Controle de Vôo (FCM, abreviatura de “Flight Controls Module”), responsável por calcular posições em comandos de vôo para correção de atitude de modo a manter a trajetória programada.
- b) Unidade de Processamento de Anemometria (ADC, abreviatura de “Air Data Computer”), responsável por adquirir e processar medidas de temperatura, pressão atmosférica estática e dinâmica para determinação de velocidade e altitude.
- c) Unidade de Posicionamento Global (GPU, abreviatura de “Global Positioning Unit”), responsável por produzir dados de velocidade e posição a partir do Sistema Global de Posicionamento (“Global Positioning System”, GPS).
- d) Unidade de Navegação Inercial (INS, abreviatura de “Inertial Navigation System”), responsável por processar acelerações lineares e angulares sentidas pelo veículo e produzir deslocamentos e velocidades lineares e angulares.
- e) Unidade Concentradora de Dados (DCU, abreviatura de “Data Concentrator Unit”), responsável pela aquisição de sinais gerados pelos comandos de vôo para a determinação de sua posição instantânea.
- f) Unidades Eletrônicas de Controle de Atuadores (ACE1 e ACE2, abreviatura de “Actuator Control Electronics”), responsáveis por acionar os comandos de vôo a partir de dados enviados pela Unidade de Controle de Vôo (FCM).

As análises conduzidas neste trabalho dependem apenas da direção, frequência e da quantidade de dados que cada unidade envia ou recebe. Portanto, não se deve esperar correção absoluta na funcionalidade atribuída a cada uma delas.

Para facilitar a compreensão da interação entre as unidades deste modelo de corretor de trajetória, é conveniente mostrar, primeiro, uma das implementações possíveis que, em geral, é chamada de “federada”, em contraposição a implementações “integradas”, como as plataformas IMA descritas nos documentos RTCA (2005).

5 ANÁLISE DO DETERMINISMO NA COMUNICAÇÃO PONTO-A-PONTO EM UMA IMPLEMENTAÇÃO FEDERADA DO CORRETOR DE TRAJETÓRIA

5.1 Introdução

A designação “federada” é um termo costumeiro na indústria aeroespacial. Ela em geral é associada a uma implementação de sistema computacional embarcado onde cada unidade desempenha uma função específica dedicada, sem preocupação com funções de outras unidades a menos de uma interface de comunicação de dados comum baseada em alguma norma da indústria, por exemplo, ARINC-429 (ARINC, 2001). Esta implementação se provou robusta ao longo dos anos porque uma falha de hardware ou software de uma unidade tem impacto bem delimitado em outras funções do sistema embarcado. Contudo, ela pode resultar em maior volume e peso, em particular de peso de cablagens, em comparação com implementações “integradas” de funcionalidade equivalente.

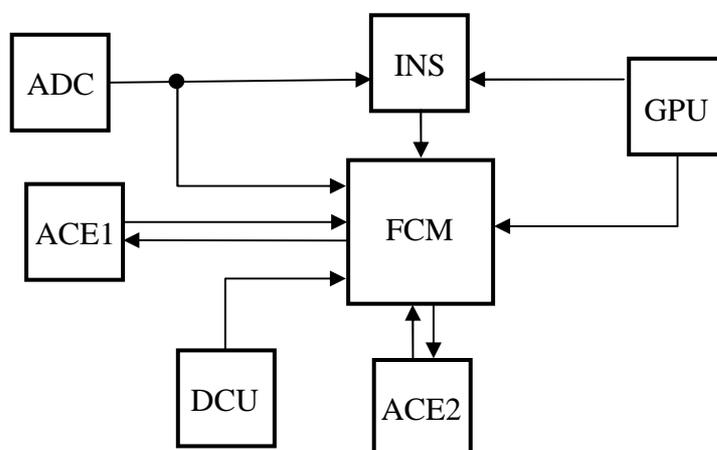


Figura 5.1 – Corretor de trajetória em implementação “federada”.

Em implementações “federadas” como a da Figura 5.1,,é comum a comunicação de dados entre unidades ser ponto-a-ponto já que a integração do sistema se realiza a partir da compatibilização das interfaces de entrada e

de saídas de dados entre uma unidade e as outras com as quais ela se relaciona funcionalmente.

A Figura 5.1 ilustra como as unidades se interligam para a transmissão e recepção de dados:

- a) A ADC se comunica com a INS e com a FCM unidirecionalmente por apenas um canal de comunicação. A representação usada (apenas uma linha originando da ADC e dividindo-se em duas, uma para a INS e outra para a FCM) é uma situação real, porque muitas vezes um cabo de comunicação de dados tem um conector de entrada e dois ou mais de saída como em implementações com ARINC-429 (ARINC, 2001). Nestes casos, dados transmitidos da origem são propagados eletricamente para todos os destinos.
- b) A GPU se comunica com a INS e com a FCM unidirecionalmente mas, ao contrário da ADC, usa dois canais de comunicação. Esta também é uma situação real porque muitas unidades que processam sinais do GPS são fornecidas com dois canais de comunicação de dados que podem ou não operar simultaneamente. Neste modelo, a GPU replica os dados em ambos os canais.
- c) A DCU se comunica com a FCM unidirecionalmente usando apenas um canal de comunicação. Este é certamente o caso trivial.
- d) As ACE1 e ACE2 se comunicam com a FCM bidirecionalmente. Em casos assim, existem duas implementações possíveis: i) a transmissão e a recepção de dados são feitas em canais de comunicação separados, caracterizando uma comunicação “full-duplex”; ii) a transmissão e a recepção de dados são feitas em um mesmo canal de comunicação, caracterizando uma comunicação “half-duplex”. Neste modelo, as comunicações com as ACE1 e ACE2 serão “full-duplex”.

- e) A FCM precisa ter múltiplos canais de comunicação, alguns com características “full-duplex” para comunicação com as ACE1 e ACE2. Lembre-se que canais de comunicação “full-duplex” em geral podem operar “half-duplex” sem qualquer prejuízo.

Em uma implementação “federada”, interações entre unidades ficam bem caracterizadas (possivelmente porque a realização física se aproxima muito da idealização lógica), facilitando a compreensão delas em outras implementações mais à frente neste trabalho.

5.2 Análise do Determinismo na Comunicação Ponto-a-Ponto

Na situação ilustrada na Figura 5.1, as unidades trocam “mensagens”, estruturas lógicas que são objeto de transmissões e recepções de dados entre elas. Normalmente, o meio físico de comunicação limita o tamanho máximo de mensagens, em geral expresso em *bits* ou *octetos* (conjuntos de oito *bits*). Neste modelo, as mensagens trocadas entre as unidades tem sua origem, destino, tamanho e frequência de transmissão listadas na Tabela 5.1.

Tabela 5.1 – Tabela de mensagens para o corretor de trajetória.

Origem	Destino	Tamanho [ud]	Frequência [s^{-1}]
ADC	INS	6	10
ADC	FCM	6	10
INS	FCM	12	40
GPU	INS	6	10
GPU	FCM	6	10
DCU	FCM	3	20
FCM	ACE1	2	20
FCM	ACE2	4	20
ACE1	FCM	2	20
ACE2	FCM	4	20

Ao longo deste trabalho, os tamanhos das mensagens trocadas entre unidades serão expressos em “unidades de dados” [ud], de modo a abstrair o formato de memória do dado transmitido (*byte*, *integer*, *float*, etc). A frequência de transmissão sempre será expressa em [s⁻¹] e velocidade de transmissão em [ud/s].

Partindo da Tabela 5.1, é fácil identificar que a unidade mais sobrecarregada em termos de entrada e saída de dados é a FCM. Na Tabela 5.2 está calculada a quantidade de dados que esta unidade recebe ou transmite por unidade de tempo.

Tabela 5.2 – Tabela de mensagens para a unidade FCM.

Direção	Tam.Mensagem [ud]	Frequência [s ⁻¹]	Taxa [ud/s]
entrando	6	10	60
entrando	12	40	480
entrando	6	10	60
entrando	3	20	60
entrando	2	20	40
entrando	4	20	80
Subtotal		120	780
saindo	2	20	40
saindo	4	20	80
Subtotal		40	120
Total Geral		160	900

Pela Tabela 5.2, a FCM precisa ser capaz de manipular 8 mensagens diferentes 160 vezes por segundo, e ser capaz de tratar 780 ud/s no canal de recepção e 120 ud/s no canal de transmissão.

Estes dados por si só garantem a correta especificação de um requisito importante para o projeto da FCM e das outras unidades: a velocidade mínima

do meio físico de comunicação em termos de unidades de dados por segundo. Neste modelo, qualquer meio físico que suporte uma taxa líquida mínima de 480 ud/s, a maior taxa da Tabela 5.2, seria adequado para todas as unidades do corretor de trajetória. Para efeito de comparação, o meio físico ARINC-429 opera com uma velocidade máxima de 100.000 bit/s (ARINC, 2001). Como cada mensagem ARINC-429 usa 32 bits para transmitir até 19 bits de dados, este meio físico transmite até 59.375 bits de dados por segundo ($=100.000 \cdot 19/32$). Se o tipo de dado transmitido for um *integer* de 32-bits, então 480 ud/s correspondem a 15.360 bits/s ($=32 \cdot 480$), resultando em uma taxa de utilização do meio físico de aproximadamente 26% ($=15360/59375$). Para ser mais preciso, neste caso hipotético de cada unidade de dado ser um *integer* de 32 bits, deve-se levar em conta que seria necessário transmitir duas mensagens ARINC-429 (38 bits de dados) para cada *integer*, o que representa 18.240 bits/s para 480 ud/s ($18240=38 \cdot 480$), elevando a taxa de utilização do meio físico para aproximadamente 31% ($=18240/59375$).

Em situações como a ilustrada na Figura 5.1, pode-se dizer que o determinismo na comunicação de dados entre unidades, cuja análise é objetivo maior deste trabalho, depende exclusivamente das implementações das operações de transmissão e recepção em cada uma delas já que o tráfego de dados entre unidades se realiza ponto-a-ponto. Quando se usa o meio físico ARINC-429, é comum implementar transmissões cíclicas onde um grupo fixo de mensagens é transmitido ordenadamente com uma frequência fixa, formando o que normalmente se convencionou chamar de “minor frame”, cuja coleção forma um “major frame”, como ilustrado na Figura 5.2. Transmissões cíclicas de tamanho e frequências fixas são determinísticas por construção: a unidade é ligada e, assim que possível, as transmissões são iniciadas e mantêm seu ritmo inalterado até a unidade ser desativada por qualquer motivo.

Quanto à recepção, cada unidade precisa ser capaz de processar as mensagens que chegam a ela na frequência e tamanhos especificados. Em geral, isto depende diretamente do hardware e do software envolvidos e se torna mais complexo se a unidade opera com um Sistema Operacional de

Tempo-Real. Um carácter determinístico na recepção de mensagens de dados é obtido se o hardware e o software envolvidos proporcionam tempos estáveis de resposta a um evento de entrada de dados. Por exemplo, a recepção de uma nova mensagem pela interface de comunicação gera uma interrupção de hardware seguida de uma interrupção de software cujos tempos podem ser medidos com precisão. Se este tempo é estável (média igual ou muito próxima da moda), pode-se dizer que, neste sistema, a recepção de mensagens é determinística.

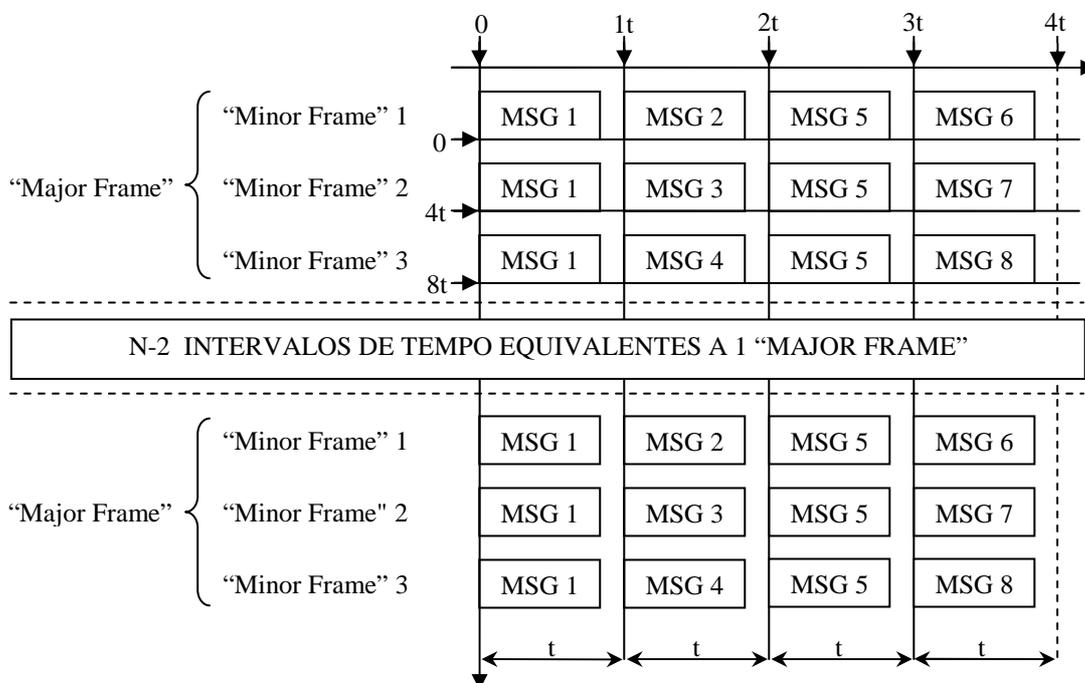


Figura 5.2 – Transmissão cíclica usando "minor frames" e "major frames".

Nos próximos capítulos serão apresentadas outras implementações mais "integradas" do corretor de trajetória nas quais serão aplicadas outras abordagens para análise do determinismo na comunicação de dados envolvida.

6 ANÁLISES DO DETERMINISMO NA COMUNICAÇÃO POR REDE EM IMPLEMENTAÇÕES INTEGRADAS DO CORRETOR DE TRAJETÓRIA

6.1 Introdução

Neste capítulo serão estudadas duas implementações “integradas” do corretor de trajetória em contraposição à implementação “federada” apresentada no Capítulo 5, distintas entre si apenas pela topologia da rede que interliga suas unidades: i) em “estrela”, usando um “switch”; ii) em “barramento”.

Em cada uma delas, as condições para um comportamento determinístico da comunicação de dados serão identificadas e os valores de parâmetros essenciais da rede serão determinados de modo a satisfazer estas condições.

6.2 Análise do Determinismo em uma Topologia em “Estrela” usando um “Switch”

A Figura 6.1 ilustra uma implementação integrada do corretor de trajetória empregando uma topologia em “estrela” que depende de um controlador central para comutar o tráfego de mensagens entre suas unidades. Este controlador central será denominado simplesmente “switch”, abstraindo o fato de ele operar com um meio físico particular, por exemplo, Ethernet. A característica de um “switch” que tem impacto neste estudo é o uso que se faz de filas de entrada ou saída ligadas às portas físicas, que podem se transformar em pontos de congestionamento de mensagens que entram ou saem do “switch”.

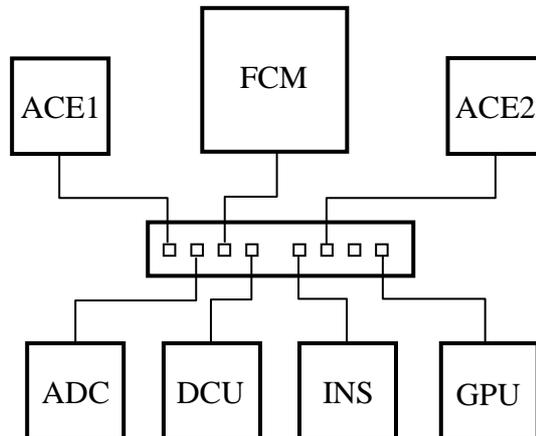


Figura 6.1 – Corretor de trajetória usando topologia “estrela”.

Neste estudo, será considerado que o “switch” opera com tecnologia “shared memory” (SEIFERT, 2000) com uma única fila de saída para cada porta física. Será considerado também que as portas físicas operam “full-duplex” e que mensagens são recebidas e encaminhadas às filas de saída passando pela memória compartilhada sem qualquer atraso. Em um caso real, este atraso depende do projeto de hardware e software do “switch”, mas em geral é uma medida estável (ALENA et al, 2006).

A unidade FCM é de especial interesse, porque ela é a unidade mais carregada em termos de entrada e saída de dados como exposto na Tabela 6.1.

A Figura 6.2 modela a porta física do “switch” ligada à unidade FCM, mostrando um “caminho livre” até alguma fila de saída passando pela memória compartilhada para mensagens que entram, e uma única fila de saída para mensagens que saem.

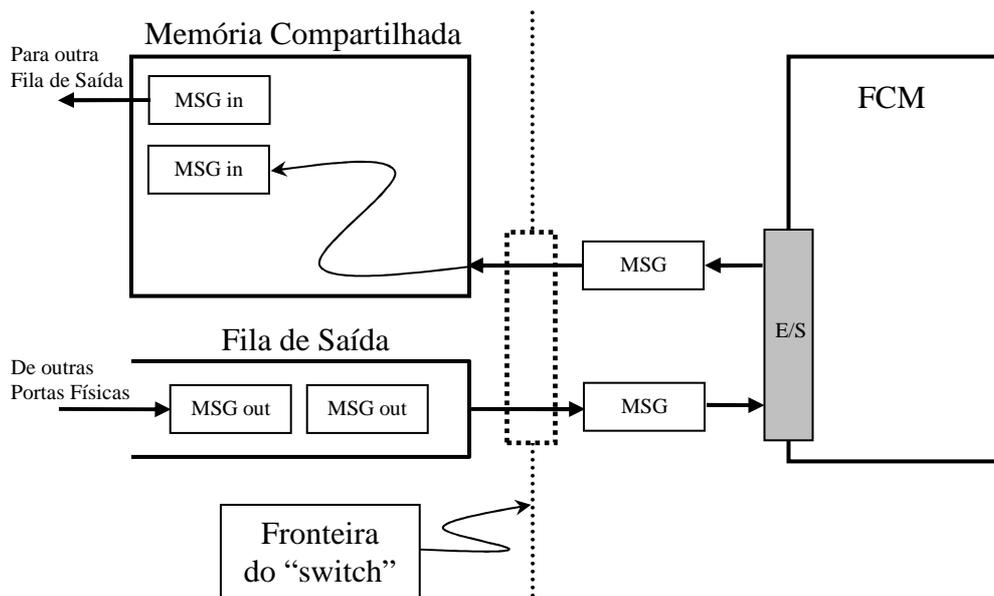


Figura 6.2 – Modelo da porta física do “switch” ligada à FCM.

A Tabela 5.2 lista as mensagens que entram e saem da FCM. Estas mesmas informações referentes apenas a mensagens que entram estão copiadas na Tabela 6.1 ordenadas pelo seu Período (inverso da Frequência) expresso em milissegundos [$\times 10^{-3}$ s].

Tabela 6.1 – Tabela de mensagens entrando na unidade FCM.

Origem	Tam.Mensagem [ud]	Frequência [s ⁻¹]	Taxa [ud/s]	Período [$\times 10^{-3}$ s]
INS	12	40	480	25
AC2	4	20	80	50
DCU	3	20	60	50
ACE1	2	20	40	50
ADC	6	10	60	100
GPU	6	10	60	100
Total	33		780	

A taxa de mensagens que sai da FCM pela Tabela 5.2 é 120 ud/s, contra 780 ud/s de mensagens que entram. Levando em conta que o “switch” modelado opera com filas de saída, vai-se nos concentrar apenas nas mensagens que saem do “switch” e entram na FCM.

Abstraindo o fato de que mensagens trocadas entre as unidades do corretor de trajetória são estruturas discretas, pode-se idealizar o fluxo de dados que entra na FCM como um fluxo contínuo de 780 ud/s. Costuma-se chamar esta idealização de “modelo fluido” (STILIADIS;VARMA, 1998). Fazendo isto, pode-se obter uma Curva de Chegada e uma Curva de Serviço para o modelo da porta física do “switch” e usar os resultados do “Network Calculus” do Capítulo 2 para determinar alguns parâmetros fundamentais desta rede em topologia “estrela”. A Curva de Chegada modela o fluxo de dados que se acumula na fila de saída e a Curva de Serviço modela o modo como a porta física do “switch” serve este fluxo que se dirige à FCM.

Para a Curva de Chegada, será usada uma função “Affine” $\alpha_{r,b}$ (Equação 2.11). Será atribuído à taxa r o valor do fluxo contínuo de 780 ud/s e à “rajada” (“burst”) b a soma dos tamanhos de todas as mensagens que transitam dentro do “switch” em direção à porta física do “switch” ligada à FCM, ou seja, 33 ud. Adotar este valor equivale a admitir que, no pior caso, todas as mensagens digitadas à FCM chegam simultaneamente à porta física do “switch” ligada a ela, uma ocorrência semelhante ao “instante crítico” descrito no Capítulo 3.

Para a Curva de Serviço, será usada uma função “Rate-Latency” $\beta_{R,T}$ (Equação 2.12). Para determinar a taxa de serviço R , será considerada que o atraso de serviço T é igual ao tempo de servir a maior mensagem de 12 ud como em Grieu (2004) e que o atraso máximo (“delay bound”), segundo limite do “Network Calculus”, não deve ser maior que o menor período 25×10^{-3} s, ou seja, que as mensagens de maior frequência 40 s^{-1} precisam ser servidas em um intervalo de tempo no máximo igual ao seu período. Esta consideração na

verdade reflete a escolha do atraso máximo (“delay bound”) na transmissão não só das mensagens de frequência 40 s^{-1} mas também de todas as outras.

Usando a expressão para o “delay bound” $T+b/R$ mostrada na Figura 2.9 e substituindo T por $12/R$, onde 12 é o tamanho da maior mensagem, temos:

$$25 \times 10^{-3} \text{ [s]} = 12[\text{ud}]/R + 33[\text{ud}]/R$$

$$R = (12+33)/25 \times 10^{-3}; R = 1800 \text{ ud/s}$$

$$T = 12/1800; \quad T = 6,66 \times 10^{-3} \text{ s}$$

Assim, a Curva de Chegada e a Curva de Serviço que modelam a porta física do “switch” ligada à FCM são as Equações 6.1 e 6.2 (o sinal “+” no lado direito da expressão entre colchetes indica que ela é válida somente quando for positiva):

$$\alpha_{r,b}(t) = 0, t = 0 \quad (6.1)$$

$$\alpha_{r,b}(t) = 780.t + 33, t > 0$$

$$\beta_{R,T}(t) = 0, 0 \leq t \leq T \quad (6.2)$$

$$\beta_{R,T}(t) = 1800[t - 6,66 \times 10^{-3}]^+$$

Portanto, o “switch” precisa ser capaz de servir o fluxo de dados que se dirige à FCM a uma taxa mínima de 1800 ud/s para atender ao requisito de atraso máximo de $25 \times 10^{-3} \text{ s}$.

A expressão $b+r.T$ do máximo tráfego acumulado (“backlog bound”), primeiro limite do “Network Calculus”, ao ser avaliada usando os valores acima resulta em 38,2 ud ($= 33+780 \times 6,66 \times 10^{-3}$). É interessante notar que este valor é maior que a soma de todas as mensagens dirigidas à FCM no intervalo de $25 \times 10^{-3} \text{ s}$ porque assumiu-se o modelo “fluido” ; e que nela, a parcela $r.T$ representa o acúmulo de tráfego no período T de atraso de serviço causado pela entrada de dados à uma taxa contínua r . Como mensagens são estruturas discretas, neste caso basta que a fila de saída ligada à porta física seja capaz de armazenar 33 ud. Filas de entrada ou saída são materializadas em hardware pela instalação de uma memória que funciona como um armazenador

intermediário (“buffer”) ligado ao circuito de transmissão de sinais pelo meio físico. Portanto, neste caso, basta que ele acomode 33 ud no pior caso.

Outro resultado pode ser obtido usando a regra simples derivada da análise feita no Capítulo 3: *a soma total dos tempos de todas as transmissões não pode ser maior que o menor período*. Neste caso, o “switch” precisa processar todo o tráfego acumulado na fila de saída no pior caso, ou seja, quando todas as mensagens chegam simultaneamente na porta física do “switch”, em um intervalo de tempo menor ou igual ao menor período. Com os dados da Tabela 6.1 (a soma de todas as mensagens é 33 ud e o menor período é 25×10^{-3} s), o “switch” precisa ser capaz de processar a fila de saída a uma taxa de 1320 ud/s ($= 33/25 \times 10^{-3}$) para satisfazer à regra do Capítulo 3. Observe-se que este resultado equivale a usar a expressão do “delay bound” fazendo o atraso de serviço T igual a zero.

Esta taxa é aproximadamente 27% menor que a taxa obtida usando os recursos do “Network Calculus”. Em termos de projeto, é conveniente ter margem na velocidade de processamento de dados do “switch”, mas ser conservador demais pode limitar, por exemplo, a escolha do tipo de meio físico adequado à rede do corretor de trajetória.

Nestes casos em que análises como as deste capítulo produzem resultados apreciavelmente diferentes, é conveniente recorrer a ferramentas de simulação ou a ensaios práticos como em Georges et al (2005) e Alena et al (2006), exercitar um número finito de casos e comparar teoria e prática.

Quanto às características essenciais da rede em topologia “estrela” e, por conseguinte, de seu “switch” controlador central, pode-se enumerar:

- 1) Cada porta física do “switch” deve acomodar no mínimo 33 ud em seu “buffer” para que não haja perdas na recepção de mensagens;

2) O meio físico deve operar a uma taxa mínima de 1.320 ud/s e a uma taxa de 1.800 ud/s no melhor caso para que todas as mensagens sejam transmitidas em, no máximo, 25×10^{-3} s.

Se as condições acima forem satisfeitas, então a comunicação de dados na rede proposta é previsível (não haverá perda de mensagens e cada uma será transmitida com uma frequência que oscila em torno da sua frequência programada dentro de um limite definido) e portanto determinística.

Convém lembrar que, por simplificação, certos tipos de atrasos não foram considerados, como, por exemplo, o tempo que uma mensagem leva desde a recepção até seu enfileiramento na porta de saída. Contudo, estes atrasos são, em geral, medidas estáveis como mostrado em Grieu (2004) e Alena et al (2006) e podem ser incluídos no termo T de atraso de serviço na expressão da Curva de Serviço ou descontados do período para cálculo da taxa de transmissão usando a regra do Capítulo 3.

6.3 Reanálise Usando Filas de Alta e Baixa Prioridade

Grieu (2004) estende seu modelo de “switch” apresentando um novo cálculo de Curvas de Serviço assumindo duas filas de saída com prioridades fixas para cada porta física: uma para mensagens de “alta prioridade” e outra para mensagens de “baixa prioridade”. Seifert (2000) defende o uso de três prioridades (“alta”, “média” e “baixa”), enquanto que alguns “switches” Ethernet comerciais operam com até 16 níveis de prioridade em cada porta física.

Grieu deriva as Curvas de Serviço usando funções “Rate-Latency” $\beta_{R,T}$ a partir do que foi apresentado por Le Boudec e Thiran (2004), admitindo que as Curvas de Chegada são limitadas por funções “Affine” $\alpha_{r,b}$. e que o protocolo de transmissão é “não-preemptivo”, consistentemente com o que foi apresentado no Capítulo 3. Para tanto, é necessário fixar a capacidade de transmissão de

dados do meio físico C , que é, por consequência, a máxima taxa com que o “switch” pode servir um fluxo de entrada.

As novas expressões para estas curvas de serviço são as Equações 6.3 e 6.4 ($\beta_{R,T}(t) = 0, 0 \leq t \leq T$):

$$\beta_{High}(t) = C.[t - L_{max}^{Low}/C]^+ \quad (6.3)$$

$$\beta_{Low}(t) = (C - r_{High}).[t - b_{High}/(C - r_{High})]^+ \quad (6.4)$$

Na Equação 6.3, o termo L_{max}^{Low}/C representa um atraso de serviço para o fluxo de alta prioridade equivalente ao tempo que se gasta para transmitir à taxa C a maior mensagem presente no fluxo de baixa prioridade L_{max}^{Low} . Na Equação 6.4, r_{High} e b_{High} são os parâmetros da Curva de Chegada “Affine” $\alpha_{r,b}$ do fluxo de alta prioridade. A diferença $(C - r_{High})$ representa a taxa líquida com que o fluxo de baixa prioridade é servido e o termo $b_{High}/(C - r_{High})$ representa um atraso de serviço correspondente ao tempo de transmissão da “rajada” (“burst”) b_{High} do fluxo de alta prioridade à taxa líquida $(C - r_{High})$.

Se partindo da Tabela 6.1 arbitrar-se que, para o corretor de trajetória, as mensagens que chegam das unidades INS, ACE1, ACE2 e DCU para a FCM tem maior prioridade, pode-se separar as mensagens em dois grupos e reorganizar a Tabela 6.1 na Tabela 6.2.

Tabela 6.2 – Mensagens entrando na unidade FCM e suas prioridades.

Origem	Prioridade	Tam.Mensagem [ud]	Frequência [s ⁻¹]	Taxa [ud/s]	Período [x10 ⁻³ s]
INS	Alta	12	40	480	25
ACE2	Alta	4	20	80	50
DCU	Alta	3	20	60	50
ACE1	Alta	2	20	40	50
ADC	Baixa	6	10	60	100
GPU	Baixa	6	10	60	100
Total		33		780	

As novas Curvas de Chegada são as funções “Affine” das Equações 6.5 e 6.6:

$$\alpha_{High}(t) = (12+4+3+2) + (480+80+60+40).t$$

$$\alpha_{High}(t) = 21 + 660.t \quad (6.5)$$

$$\alpha_{Low}(t) = (6+6) + (60+60).t$$

$$\alpha_{Low}(t) = 12 + 120.t \quad (6.6)$$

Para derivar novas Curvas de Serviço, a taxa C será fixada em 1.800 ud/s como na Equação 6.2. Assim, as novas Curvas de Serviço são as funções “Rate-Latency” das Equações 6.7 e 6.8:

$$\beta_{High}(t) = 1800.[t - 6/1800]^+$$

$$\beta_{High}(t) = 1800.[t - 3,33 \times 10^{-3}]^+ \quad (6.7)$$

$$\beta_{Low}(t) = (1800 - 660).[t - 21/(1800 - 660)]^+$$

$$\beta_{Low}(t) = 1140.[t - 18,4 \times 10^{-3}]^+ \quad (6.8)$$

Para o fluxo de alta prioridade, o “backlog bound” $b+r.T$ resulta em 23,2 ud (= $21 + 660 \times 3,33 \times 10^{-3}$) e o “delay bound” $T+b/R$ resulta em 15×10^{-3} s (= $3,33 \times 10^{-3} + 21/1800$). Para o fluxo de baixa prioridade, o “backlog bound” resulta em 14,2 ud (= $12 + 120 \times 18,4 \times 10^{-3}$) e o “delay bound” em $28,9 \times 10^{-3}$ s (= $18,4 \times 10^{-3} + 12/1140$). Para o fluxo de alta prioridade, a razão entre o “backlog bound” e a soma dos tamanhos das mensagens é menor que a obtida

a partir das Equações 6.1 e 6.2 ($23,2/21 < 38,2/33$), enquanto que para o fluxo de baixa prioridade é maior ($38,2/33 < 14,2/12$). Um resultado curioso é que soma dos “backlog bounds” ($37,4 = 23,2 + 14,2$) é menor que o “backlog bound” obtido a partir das Equações 6.1 e 6.2 ($38,2$). Para o fluxo de alta prioridade, o “delay bound” é 40% melhor ($1 - 15/25$) e para o de baixa prioridade é 16% pior ($28,9/25 - 1$).

A adoção de duas filas de prioridades para a porta física do “switch” melhora um aspecto e piora outro; portanto cabe ao projetista julgar que implementação se adequa melhor à situação à sua frente.

6.4 Análise do Determinismo em uma Topologia em “Barramento” usando “Time-Division Multiple Access” (TDMA)

A Figura 6.3 ilustra uma implementação integrada do corretor de trajetória empregando uma topologia em “barramento”. O mais importante nesta representação é notar que o meio físico de transmissão aparece como uma entidade passiva destacada das unidades que o compartilham, ao contrário da implementação anterior em “estrela” que dependia de uma entidade ativa (um controlador central do tipo “switch”).

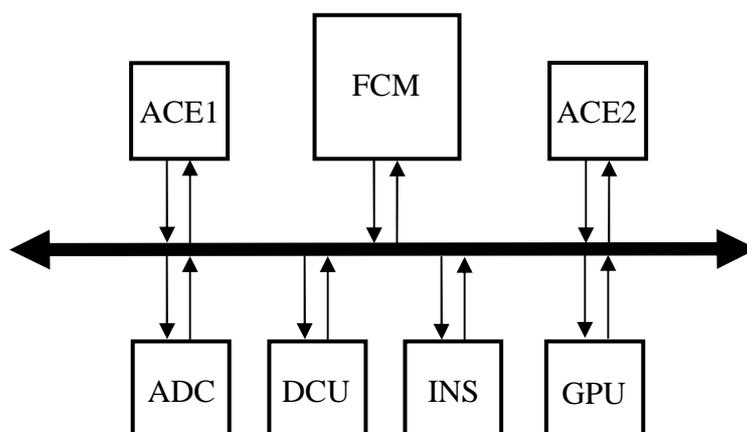


Figura 6.3 – Corretor de trajetória usando topologia em “barramento”.

A implementação mais popular desta topologia materializou-se em redes Ethernet desenvolvidas pela Xerox, Digital Equipment Corporation (DEC) e Intel durante a década de 1980 (rever Capítulo 2). A partir de 1991, uma outra implementação originada na indústria automobilística, chamada “Controller Area Network” (CAN) e desenvolvida pela Robert Bosch alemã (CAN, 1991), também se tornou bastante popular.

Tanto Ethernet como CAN oferecem às unidades ligadas em rede um meio físico de transmissão compartilhado, em geral um cabo metálico terminado resistivamente em ambas as pontas. Em ambas, a comunicação entre uma unidade qualquer e todas as outras ocorre em modo “broadcast” (uma unidade

transmite e todas as outras recebem) e em modo “half-duplex” (uma unidade ou transmite ou recebe, nunca transmite e recebe simultaneamente). Contudo, cada implementação adotou uma estratégia diferente para lidar com conflitos de acesso ao meio físico. Este conflito ocorre quando duas unidades quaisquer, ao perceber o meio físico desimpedido (em geral esta detecção é feita pela presença ou ausência de uma onda portadora), iniciam simultaneamente uma tentativa de transmissão. No caso Ethernet, seus projetistas elaboraram o que foi chamado de “back-off mechanism”: cada unidade ao perceber outra transmissão em andamento, interrompe a sua própria e aguarda um período aleatório para iniciar uma nova tentativa. No caso CAN, seus projetistas impuseram uma prioridade de transmissão intrínseca ao identificador da mensagem transmitida: quanto menor o número do identificador, maior prioridade tem a mensagem. Unidades ligadas em rede CAN percebem uma transmissão de maior prioridade comparando o nível lógico presente no meio físico com cada bit transmitido: se eles forem desiguais, uma mensagem de maior prioridade está sendo transmitida e a unidade interrompe sua transmissão imediatamente. Uma vantagem da rede CAN é que, em caso de conflito, a mensagem de maior prioridade sempre continua sua transmissão sem interrupção.

Em redes Ethernet com topologia “barramento”, não é possível determinar *a priori* limites de atrasos de transmissão apenas olhando as características de cada mensagem a ser transmitida pelo simples fato de que não há nesta implementação algo que imponha um ordenamento ou uma prioridade nas transmissões de mensagens. Por conta disto, costuma-se dizer que a transmissão de mensagens em rede Ethernet acontece em regime “best effort”. Em redes CAN, é possível determinar um limite superior do atraso de transmissão de uma determinada mensagem somando o tempo de transmissão de todas as outras mensagens de maior prioridade com as quais há certeza de conflito de acesso ao meio físico compartilhado.

Neste estudo, será utilizada uma outra estratégia desenvolvida mais recentemente e que pertence a uma família de protocolos denominados de “time-triggered” (disparado por tempo) para acesso a um meio físico compartilhado que opera em modos “broadcast” e “half-duplex” (TTP/C, 2003). Parte desta estratégia foi apresentada no Capítulo 4 (transmissão cíclica usando “minor frames e “major frames” em meio físico ARINC-429).

A estratégia se baseia em uma técnica conhecida no mundo das telecomunicações por “Time Division Multiple Access” (TDMA). Por TDMA entende-se permitir a unidades interligadas por um meio físico compartilhado usar toda sua capacidade de transmissão por um período limitado de tempo, uma de cada vez (ANSWERS, 2008). Em geral, divide-se o tempo em “TDMA slots” (“casas”), isto é, intervalos de igual duração, atribui-se a cada unidade um certo número deles e sua vez na sequência de transmissão. Esta atribuição pode ser feita estática ou dinamicamente. Em ambientes embarcados, costuma-se dar preferência para atribuições estáticas definidas em tempo de projeto, o que assegura o caráter determinístico de transmissões de mensagens usando TDMA.

Na família de protocolos “time-triggered”, é comum referir-se a uma sequência repetitiva de “TDMA slots” como “TDMA round” (“rodada”). Neste estudo, chamar uma sequência repetitiva de “TDMA rounds” será chamada de “TDMA cycle” (“ciclo”). No Capítulo 4, uma sequência cíclica de transmissões de mensagens em meio físico ARINC-429 foi dividida em “minor frames” e “major frames” como uma forma de multiplexação no tempo. Em TDMA, a idéia é exatamente a mesma.

A Tabela 6.1 lista todas as mensagens trocadas entre as unidades do corretor de trajetória. Lembrando que as unidades ADC e GPU transmitem mensagens replicadas simultaneamente para as unidades INS e FCM, pode-se refazer a Tabela 6.1 na Tabela 6.3. Esta resume todas as mensagens que precisam ser transmitidas no meio físico compartilhado desta rede em topologia

“barramento” e acrescenta uma informação importante para a estratégia TDMA deste estudo: a frequência relativa, ou seja, a razão entre uma frequência qualquer e o Máximo Divisor Comum (MDC) das frequências do conjunto de mensagens analisado.

Tabela 6.3 – Tabela de mensagens e suas frequências relativas.

Origem	Tamanho [ud]	Frequência [s⁻¹]	Freq. Relativa
INS	12	40	4
FCM	4	20	2
ACE2	4	20	2
DCU	3	20	2
FCM	2	20	2
ACE1	2	20	2
ADC	6	10	1
GPU	6	10	1

O que será feito a seguir é usar uma abordagem simples para obter uma sequência de transmissões em TDMA para a rede do corretor de trajetória que assegure a cada unidade um número de casas TDMA suficiente para transmitir suas mensagens sem interrupção, sem atrasos e na frequência programada.

Se as frequências relativas são potências de 2, para montar o ciclo TDMA para o corretor de trajetória pode-se usar as seguintes regras:

1. O número de rodadas no ciclo deve ser igual à maior frequência relativa;
2. O número de casas por rodada deve ser estimado dividindo-se a soma dos produtos “Tamanho x Freq. Relativa” pelo número de rodadas arredondado ao próximo inteiro;
3. O número de casas por ciclo é igual ao número de rodadas multiplicado pelo número de casas por rodada;

4. O tempo de uma casa em segundos é igual ao inverso do número obtido pela multiplicação do número de casas por ciclo pelo valor da menor frequência do conjunto de mensagens analisado.

Com os dados da Tabela 6.3, pode-se montar a Tabela 6.4 e, com ela, dimensionar o ciclo TDMA usando as regras acima.

Tabela 6.4 – Número total de casas TDMA obtido a partir da Tabela 6.3.

Origem	Tamanho [ud]	Frequência [s⁻¹]	Freq. Relativa	Casas
INS	12	40	4	48
FCM	4	20	2	8
ACE2	4	20	2	8
DCU	3	20	2	6
FCM	2	20	2	4
ACE1	2	20	2	4
ADC	6	10	1	6
GPU	6	10	1	6
			Total	90

Partindo da Tabela 6.4, temos:

1. O número de rodadas deve ser 4;
2. O número de casas é o inteiro superior a $90/4$ ($= 22,5$), portanto 23;
3. O número de casas por ciclo é 92 ($= 4 \times 23$);
4. Cada casa tem a duração de $1,087 \times 10^{-3}$ segundos (inverso de 92×10).

Até agora tem-se o que se pode chamar de “geometria” do ciclo TDMA para o corretor de trajetória, ou seja, seu formato em termos de rodadas e casas (Figura 6.4); mas nada ainda foi obtido sobre a sequência de transmissão de mensagens que as unidades devem obedecer.

rodada		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
↓ 1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
↓ 2		24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
3		47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
4		70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92
casa →		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Figura 6.4 – Formato do ciclo TDMA para o corretor de trajetória.

Para obter uma sequência (podem existir outras), será preciso criar um outro número que será chamado de “Fator de Multiplexação” que será expresso por uma fração cujo numerador é a frequência relativa da mensagem e o denominador é a maior frequência relativa do conjunto analisado. Os Fatores de Multiplexação para as mensagens da Tabela 6.4 estão indicados na Tabela 6.5 (as frações foram simplificadas).

Tabela 6.5 – Fatores de Multiplexação para as mensagens da Tabela 6.4.

Origem	Tamanho [ud]	Freq. Relativa	Fator de Multiplexação
INS	12	4	1/1
FCM	4	2	1/2
ACE2	4	2	1/2
DCU	3	2	1/2
FCM	2	2	1/2
ACE1	2	2	1/2
ADC	6	1	1/4
GPU	6	1	1/4

Usando os Fatores de Multiplexação, pode-se obter uma sequência de transmissão usando as seguintes regras simples:

1. Cada mensagem aparece em cada rodada do ciclo TDMA na medida do seu Fator de Multiplexação (no caso da Tabela 6.5, uma vez a cada rodada, uma vez a cada duas rodadas, uma vez a cada quatro rodadas, etc);

2. Uma mensagem que aparece em diferentes rodadas deve iniciar sempre na mesma posição relativa ao início de cada rodada;
3. As mensagens devem ser alocadas em rodada começando pela de Fator de Multiplexação 1 e prosseguindo na ordem decrescente dos seus Fatores de Multiplexação.

Usando estas regras, uma sequência possível para o ciclo TDMA criado a partir das mensagens da Tabela 6.3 está desenhada na Figura 6.5.

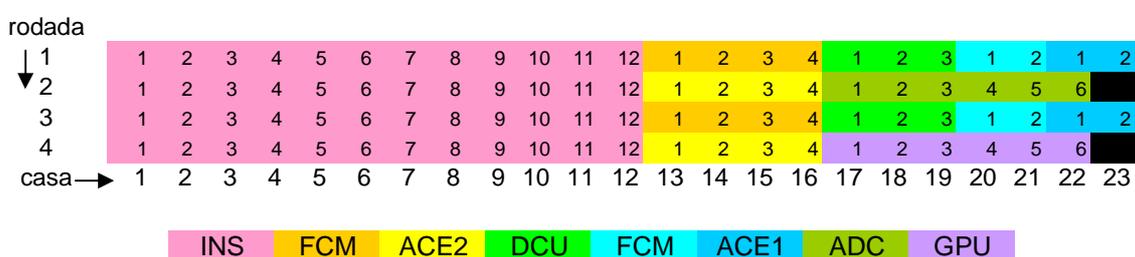


Figura 6.5 – Sequência de transmissão TDMA para o corretor de trajetória.

Notar que na Figura 6.5 as casas em cor negra representam duas casas livres, ou seja, não ocupados por transmissões de mensagens de qualquer unidade do corretor de trajetória, resultado do arredondamento feito no número de casas por rodada (92 casas por ciclo contra o total de 90 como na Tabela 6.4).

É fácil notar que a partir desta sequência outras podem ser obtidas (exemplos: trocar de posição as mensagens de tamanho 4 referentes às unidades FCM e ACE2, trocar de posição as casas 1 e 2, 3 e 4 a partir da casa 17, etc). Qual sequência faz mais sentido, cabe ao projetista do sistema decidir.

Este processo pode ser parcialmente automatizado se for usado um “solver” para obter a porção não-trivial do ciclo TDMA referente à sequência de transmissão de mensagens com Fator de Multiplexação menor que 1 (na Figura 6.5, casas 13 ao 23 em cada rodada). Ao usar um “solver”, a estratégia é minimizar o número de casaas que sobram.

Para transmitir mensagens em TDMA e evitar conflitos de acesso ao meio físico de transmissão compartilhado, cada unidade envolvida precisa: i) conhecer antecipadamente a sequência de transmissão; ii) conhecer o protocolo que determina o início do ciclo; iii) saber estimar e corrigir o erro do seu próprio relógio interno em relação aos das demais unidades. Em telefonia celular, as estações rádio-base negociam instantaneamente com cada terminal um canal de transmissão para atendimento da chamada pendente. Em sistemas integrados que operam usando protocolo TTP (TTA-GROUP, 2003), a sequência de transmissão das mensagens e outras informações relevantes são carregadas estaticamente em cada unidade antes do início da operação. O início da operação de um sistema assim obedece a uma sequência de eventos que arbitra o início do ciclo e culmina com a perfeita composição temporal das transmissões de mensagens, estabelecendo um caráter rigorosamente determinístico na comunicação entre as unidades do sistema. A correção do relógio interno para evitar conflitos de acesso ao meio compartilhado em alguns sistemas que usam protocolos “time-triggered” é feito usando o algoritmo Welch-Lynch (1988), assumindo que o desvio acumulado por conta da imprecisão dos osciladores que atualizam os relógios internos (“drift”) é pequeno e não difere muito entre uma unidade e outra. Uma explicação simplificada deste protocolo é apresentada mais adiante.

O ciclo TDMA obtido na Figura 6.5 dura exatamente 100×10^{-3} s ($= 92 \times 1,087 \times 10^{-3}$ s), portanto as mensagens com frequência relativa 4 ocorrem a cada 25×10^{-3} s, as com frequência relativa 2 a cada 50×10^{-3} s e as com frequência relativa 1 a cada 100×10^{-3} s, exatamente como especificado na Tabela 6.4 invertendo a frequência para obter o período. Nesta sequência de transmissão, a unidade INS transmite por 12 casas a partir da casa 1 das rodadas 1 a 4, a unidade FCM transmite por 4 casas a partir da casa 13 e por 2 casas a partir da casa 20 das rodadas 1 e 3, e assim por diante. Convém lembrar que admite-se que as transmissões ocorrem em modo “broadcast” e “half-duplex”

em um meio físico compartilhado, portanto uma mensagem transmitida por uma unidade é recebida simultaneamente por todas as outras.

Propositalmente, uma casa neste estudo transmite exatamente uma “unidade de dado” [ud], portanto a capacidade de transmissão do meio físico deve ser de 920 ud/s no mínimo (comparar com outros resultados na Seção 6.1). Na vida real, uma casa TDMA corresponde ao tempo necessário para transmissão de uma mensagem de comprimento mínimo na velocidade do meio físico de transmissão (no caso TTP, 240 bytes e 5 megabit/segundo em meio físico RS-485). Como em TDMA cada unidade conhece precisamente o instante de iniciar e o instante de interromper sua transmissão, só há necessidade de manter “buffers” nas interfaces de hardware com o meio físico de transmissão com tamanho suficiente para acomodar suas próprias mensagens além daquelas transmitidas por outras unidades que são necessárias à sua operação. Pela mesma razão, não se admite atrasos nas transmissões de mensagens: ou elas ocorrem ou não ocorrem. Uma ausência de transmissão no instante apropriado é considerada como uma falha a ser tratada por um protocolo adequado.

Em resumo, transmissões em regime TDMA são determinísticas por construção (são perfeitamente previsíveis e invariantes no tempo) e quando não dependem de um controlador central para arbitrar sequências de transmissão de mensagens, necessitam de outros protocolos e artifícios que assegurem uma perfeita composição temporal destas transmissões durante a inicialização do sistema e um regime permanente de correção de imprecisão de relógios internos dentro de uma tolerância pequena, em geral da ordem de frações de microssegundos.

Em sistemas embarcados críticos, onde uma falha pode ocasionar perda do veículo e de seus tripulantes, ciclos TDMA são definidos durante a fase de projeto e sua programação faz parte do conjunto de informações estáticas carregadas nas unidades que compõe o sistema durante sua integração. Por

conta disto, alterações ou extensões uma vez que o sistema entra em operação são acomodadas com maior dificuldade do que em outras implementações. Costuma-se dizer que em TDMA “não há folgas” (“no hidden reserves”), uma vez que o ciclo é projetado exatamente para um conjunto definido de mensagens. Se folgas são essenciais por demanda do projeto, elas precisariam ser introduzidas artificialmente no ciclo TDMA, possivelmente como mensagens que seriam transmitidas por unidades ainda não integradas ao sistema.

6.5 Algoritmo Welch-Lynch de Correção de Relógios em Sistemas Distribuídos

Em 1988 J. L. Welch e N. Lynch (1988) publicaram um algoritmo tolerante a falhas para correção de relógios em sistemas distribuídos, assumindo que unidades estejam interligadas em rede por um meio de transmissão confiável em regime “broadcast” e que seus relógios estejam aproximadamente sincronizados no início da operação e apresentem um desvio cumulativo (“drift”) que cresce de maneira lenta. Este algoritmo será reproduzido aqui de maneira simplificada. Outros textos mais didáticos podem ser encontrados na literatura (DUTERTRE, 1998).

A Figura 6.6 ilustra uma situação com três unidades A, B e C que transmitem três mensagens em um ciclo. Na figura, a linha horizontal representa a linha do tempo como percebida por cada unidade (a unidade de medida de tempo é irrelevante).

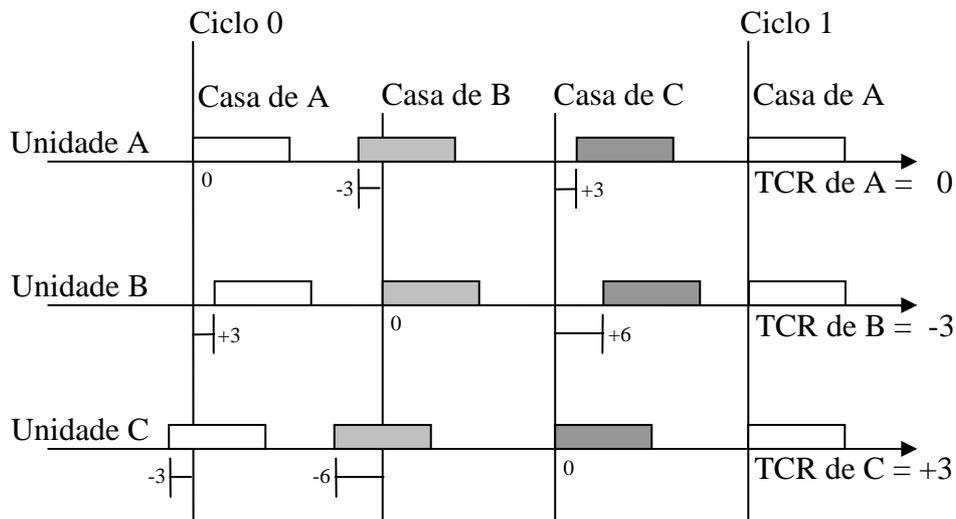


Figura 6.6 – Exemplo simplificado de aplicação do algoritmo Welch-Lynch.

A unidade A inicia o ciclo e transmite sua mensagem no tempo 0 do seu relógio interno. Esta transmissão é percebida respectivamente pelas unidades B e C com um atraso de +3 e um adiantamento de -3 em relação ao tempo conhecido por ambos para a transmissão da mensagem de A. Em seguida, a unidade B transmite no tempo 0 do seu relógio interno e esta transmissão é percebida pelas unidades A e C respectivamente com adiantamentos de -3 e -6. Em seguida, a unidade C transmite no tempo 0 do seu relógio interno e esta transmissão é percebida pelas unidades A e B respectivamente com atrasos de +3 e +6.

De posse destas medidas, cada unidade calcula seu próprio “Termo de Correção do Relógio” (TCR) usando a média aritmética de um conjunto de valores formado pelos desvios percebidos nas transmissões das outras unidades e 0 (zero), representando o desvio da unidade em relação a si própria. Assim, a unidade A calcula a média em {0, -3, +3} e obtém 0 (A não precisa corrigir seu relógio neste ciclo), a unidade B calcula a média em {+3, 0, +6} e obtém +3 (está adiantado em relação a A e C, portanto o TCR deverá ser -3), e a unidade C calcula a média em {-3, -6, 0} e obtém -3 (está atrasado em

relação a A e B, portanto o TCR deverá ser +3). Não é difícil notar que os desvios assumidos inicialmente para a construção da Figura 6.6 eram +3 para a unidade B e -3 para a unidade C.

No algoritmo original, a média dos desvios para obtenção do termo de correção é feita de uma maneira um pouco mais elaborada e, em implementações reais, as correções podem não ser feitas integralmente no ciclo seguinte (TTA-GROUP, 2003).

6.6 Revisão da Análise do Determinismo em uma Topologia em “Barramento” Usando um Modelo “Produtor-Consumidor”

A análise da Seção 6.3 anterior se preocupava exclusivamente com as transmissões de mensagens em um meio físico compartilhado, para o que foi proposta uma solução usando “Time Division Multiple Access” (TDMA) de modo a assegurar o caráter determinístico na comunicação de dados pela rede do corretor de trajetória. Nesta seção, será feito um outro exercício levando em conta aspectos que envolvem não apenas a realização da comunicação de dados entre uma unidade e outra, mas também como estes dados são aproveitados por cada unidade no desempenho de sua função específica.

Em sistemas distribuídos ou não, funções específicas como, por exemplo, a coleta de dados de sensores de atitude são implementadas parte em hardware, parte em software, na medida da conveniência do projeto levando em conta diversos fatores tais como volume, peso e custo total da solução. Com a evolução de processadores e alguns periféricos importantes (por exemplo, memória e barramentos seriais), tem sido possível implementar em software funções cada vez mais complexas usando um hardware cada vez mais padronizado.

O modelo do corretor de trajetória, como foi apresentado até agora, se compõe de unidades autônomas ligadas umas às outras por meio de uma rede de comunicação de dados de uma ou outra topologia. Em outras implementações, duas ou mais unidades são integradas em um único módulo com vantagens de redução de volume e peso, ao custo em geral do aumento na complexidade do software associado. Para esta seção, o modelo do corretor de trajetória será refeito buscando um outro nível de integração além da comunicação de dados por rede.

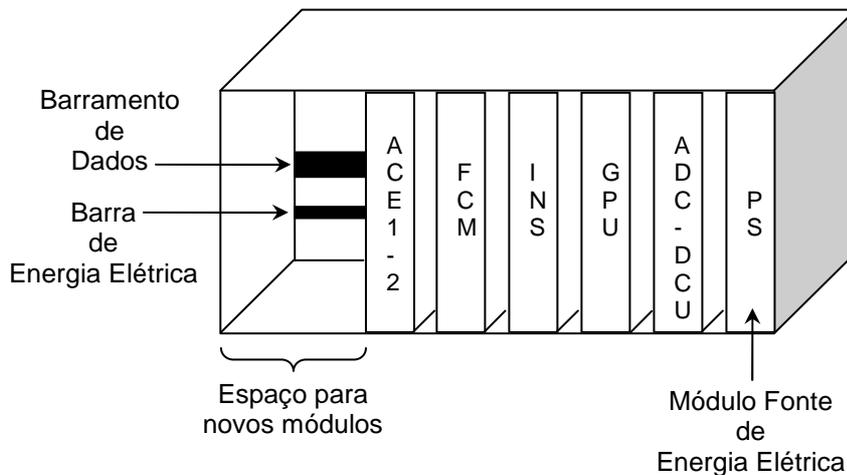


Figura 6.7 – Nova implementação do corretor de trajetória.

A Figura 6.7 ilustra uma nova implementação de hardware para o corretor de trajetória em que as unidades, em lugar de serem autônomas, são módulos de formato padronizado inseridos em um bastidor, compartilhando um barramento paralelo dedicado à transmissão de dados e uma barra de energia elétrica suprida por um módulo especial (PS na Figura 6.7). Notar que as unidades ACE1 e ACE2 e as unidades ADC e DCU foram integradas em dois módulos, respectivamente ACE1-2 e ADC-DCU. Notar também que o arranjo em bastidor não muda a essência da implementação distribuída usando uma rede em topologia “barramento” apresentada na seção anterior: a rigor, estaria-se trocando um cabo metálico aberto com terminações resistivas por filetes de um circuito impresso.

Em construções assim é possível reservar espaço para instalação de novos módulos sem aumento de volume do hardware instalado e a substituição de módulos durante a vida em serviço do sistema torna-se uma operação mais simples. Nos casos em que um único módulo abriga duas ou mais funções distintas que precisam operar com confiabilidade à despeito de falhas de uma ou outra, há que se considerar um aumento significativo de complexidade do software.

Na implementação da Figura 6.7, uma situação semelhante à que foi coberta na Seção 6.1 rodada a se manifestar. Como as unidades ACE1 e ACE2 e ADC e DCU agora são apenas dois módulos com apenas uma conexão cada ao barramento de dados do bastidor, o desempenho das funções específicas aparentemente poderá provocar um congestionamento de mensagens nesta única porta de saída de dados. Com isto, a análise feita anteriormente precisaria ser novamente aplicada para que fossem obtidos os valores mínimos de velocidade de transmissão do meio físico compartilhado e os valores mínimos de “buffers” associados a portas de saída uma vez estabelecidos os valores máximos aceitáveis de atrasos de transmissão.

Contudo, uma outra abordagem que leva em conta as interações entre as funções específicas desempenhadas por cada módulo do corretor de trajetória ajudará a encontrar a melhor maneira de lidar com a nova situação apresentada.

Não é difícil reconhecer que qualquer função desempenhada pelos módulos do corretor de trajetória depende de uma entrada de dados, de um processamento e de uma saída de dados. Assim, quanto à movimentação de dados, os módulos que juntos operam a função maior “Correção de Trajetória” atuam tanto como “consumidores” quanto “produtores”. Se uma função específica for chamada de “tarefa” e a dependência que cada tarefa tem de dados produzidos pelas outras for observada, é possível estabelecer uma sequência de “consumo” e “produção” de dados independente de como as tarefas estão alocadas nos módulos que compartilham o bastidor da Figura 6.7.

Na Figura 6.8, a tarefa Tadc é responsável por adquirir e processar medidas de temperatura, pressão atmosférica estática e dinâmica para determinação de velocidade e altitude, a tarefa Tgpu é responsável por produzir dados de velocidade e posição a partir do sistema GPS, e assim por diante como descrito originalmente no Capítulo 4 para cada uma das unidades do corretor de trajetória. Seguindo a linha do tempo e obedecendo uma relação

estritamente causal, a tarefa Tadc “produz” dados para as tarefas Tins e Tfcu, as tarefas Tins e Tdcu “produzem” dados para a tarefa Tfcu, que “produz” dados para as tarefas Tace1 e Tace2, que por fim “produzem” dados de volta para a tarefa Tfcu.

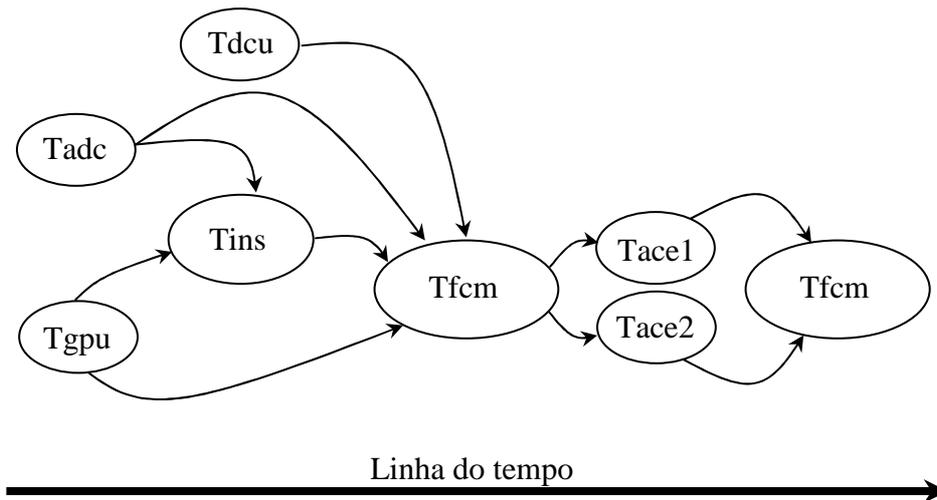


Figura 6.8 – Produção e consumo de dados no corretor de trajetória.

Usando a mesma idéia da Seção 6.2, será assumida uma estratégia TDMA de transmissão e dispor a sequência de consumo (entrada) e produção (saída) de dados em uma “rodada” (ou “round”) de 25×10^{-3} s, lembrando que este é o período mais curto entre as transmissões de mensagens na rede do corretor de trajetória como indicado na Tabela 5.1 do Capítulo 4 para a mensagem de 12du transmitida pela unidade INS para a unidade FCM como uma frequência de 40 s^{-1} . Com isto, o tempo que as tarefas do corretor de trajetória têm para processar dados será limitado de uma maneira indireta.

Na Figura 6.9, foi assumido que o meio físico de transmissão compartilhado opera com uma velocidade tal que cada módulo pode transmitir até 6 ud em apenas 1×10^{-3} s, resultando em uma capacidade de transmissão de 6.000 ud/s (comparar com os outros resultados nas Seções 6.2 – 920 ud/s – e 6.1 – 1.320 ud/s e 1.800 ud/s), e que os modos de operação “broadcast” e “half-duplex”

também valem para esta implementação em bastidor do meio físico de transmissão.

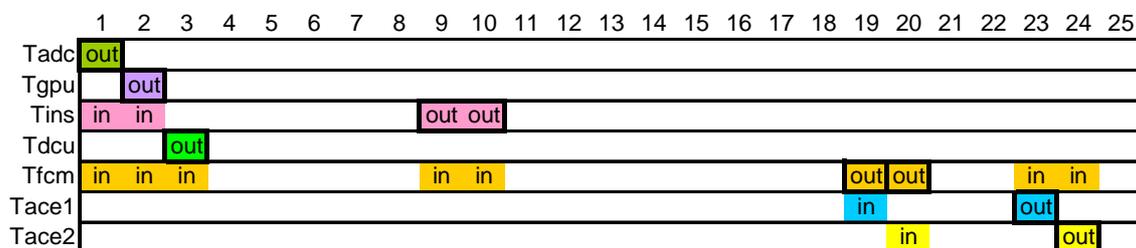


Figura 6.9 – Uma “rodada” (ou “round”) de produção e consumo de dados.

Se uma “casa” (ou “slot”) representar uma fração de 1×10^{-3} s de uma rodada, a tarefa Tadc produz dados (“out”) para consumo (“in”) das tarefas Tins e Tfcm na casa 1, a tarefa Tgpu produz dados para consumo das tarefas Tins e Tfcm na casa 2, a tarefa Tdcu produz dados para consumo da tarefa Tfcm na casa 3, a tarefa Tins produz dados para consumo da tarefa Tfcm nas casas 9 e 10, a tarefa Tfcm produz dados para consumo das tarefas Tace1 e Tace2 nas casas 19 e 20, e as tarefas Tace1 e Tace2 produzem dados para consumo da tarefa Tfcm nas casas 23 e 24. Notar que a tarefa Tins necessita de duas casas para enviar uma mensagem de 12 ud para a tarefa Tfcm (rever a Tabela 5.1 do Capítulo 5).

Esta disposição temporal, construída usando a mesma estratégia TDMA de transmissões de mensagens apresentado na Seção 6.2, atende à sequência estabelecida na Figura 6.8 e evita conflitos de acesso ao meio físico de transmissão compartilhado, mesmo considerando que as tarefas Tadc e Tdcu estão alocadas em um único módulo ADC-DCU e que as tarefas Tace1 e Tace2 estão alocadas em um único módulo ACE1-2.

Desta maneira, pode-se dizer que no que se refere à entrada e saída de dados, o caráter determinístico pode ser assegurado nesta nova implementação do corretor de trajetória desde que os módulos: i) conheçam antecipadamente a

sequência de transmissão de dados; ii) conheçam o protocolo que determina o início do ciclo; iii) saibam estimar e corrigir o erro do seu próprio relógio interno em relação aos dos demais módulos.

Lembrando que uma tarefa depende não só de entrada ou saída de dados, resta analisar se o processamento destes dados pode ser acomodado dentro do tempo arbitrado para uma rodada.

Em projetos de software para sistemas embarcados, o tempo que cada tarefa leva para executar no pior caso (“Worst Case Execution Time” – WCET) é uma informação extremamente importante e também uma fonte de incertezas. Estimativas de WCET podem ser obtidas por medição ou análise, contudo é difícil afirmar que o pior caso foi realmente observado quando se faz medições e é necessário ter componentes do processador (“instruction caching” e “pipelining”, por exemplo) adequadamente modelados quando se opta por fazer análise (BURNS;WELLINGS, 1997). Como neste estudo do corretor de trajetória não há código executável a analisar ou medir, o que será feito é examinar os limites de tempo que as tarefas tem para processar dados antes de serem novamente agendadas.

Para acomodar todas as transmissões de dados respeitando as frequências estabelecidas na Tabela 5.1 do Capítulo 5 (10, 20 e 40 s⁻¹), a partir da Figura 6.9 será composto um ciclo de 100x10⁻³ s adicionando outras três rodadas de 25x10⁻³ s. Notar que 100x10⁻³ s é o Mínimo Múltiplo Comum dos três períodos 100, 50 e 25x10⁻³ s referentes às frequências de transmissão da Tabela 6.1.

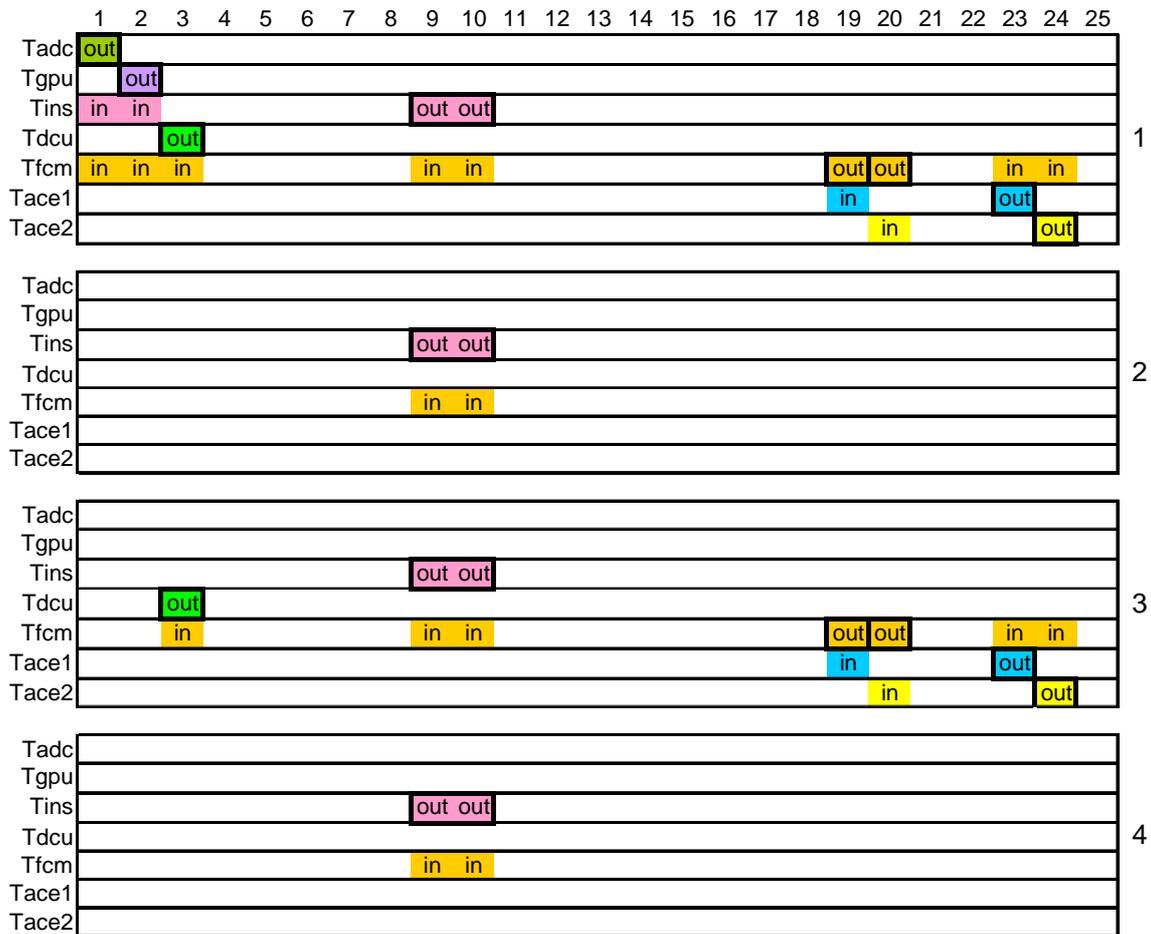


Figura 6.10 – Um ciclo completo de produção e consumo de dados.

Na Figura 6.10, as tarefas Tadc, Tgpu e Tdcu trabalham coletando dados de sensores (rever Capítulo 4) e tem respectivamente 100, 100 e 50×10^{-3} s para terminar seu processamento antes da próxima transmissão de dados para as tarefas Tins e Tfcm no próximo ciclo, admitindo hipoteticamente que o limite do tempo de processamento em cada uma delas é igual ao período.

A Figura 6.11 ilustra com mais detalhes os tempos de processamento disponíveis para as demais tarefas na primeira rodada do ciclo: nas casas em fundo cinza ocorrem atividades de entrada e saída de dados e as casas numeradas contam intervalos de 1×10^{-3} s.

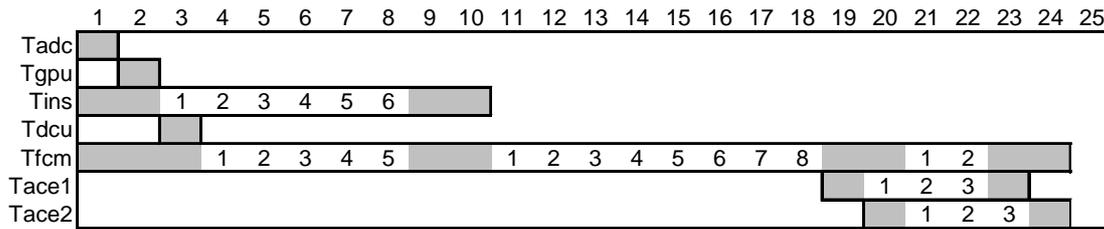


Figura 6.11 – Tempos de processamento na primeira rodada do ciclo.

A tarefa Tins, que também trabalha coletando dados de sensores e depende de dados produzidos pelas tarefas Tadc e Tgpu, tem apenas 6×10^{-3} s para processamento antes de transmitir dados para a tarefa Tfcm. As tarefas Tace1 e Tace2 têm apenas 3×10^{-3} s entre receber dados da tarefa Tfcm e transmitir uma resposta de volta. A tarefa Tfcm tem apenas 8×10^{-3} s para processamento entre receber dados da tarefa Tins e enviar dados para a tarefa Tace1. Supondo que a tarefa Tfcm faz o que se costuma chamar de “fusão de dados” (combinação de dados de mais de uma fonte para atingir maior precisão em algum cálculo), este tempo parece exíguo. Se esta fosse uma situação de fato, a conclusão poderia ser que nestas condições dificilmente o corretor de trajetória cumpriria sua função. Como alternativa imediata, sobraria aumentar a velocidade de processamento no módulo FCM (aumentar a velocidade de transmissão de dados ganharia apenas umas poucas frações de segundos), em outras palavras, uma indesejável alteração do projeto de hardware.

Outra alternativa seria rearranjar a sequência de transmissões de dados na tentativa de redistribuir preciosas frações de segundos sem desprezitar a sequência de produção de consumo de dados da Figura 6.8.

Na Figura 6.12, as transmissões de dados das tarefas Tace1 e Tace2 para a tarefa Tfcm foram deslocadas para as casas 1 e 2 da segunda rodada e as transmissões de dados da tarefa Tfcm para as tarefas Tace1 e Tace2 foram deslocadas para as casas 22 e 23 da primeira rodada, ganhando 3×10^{-3} s extras para a tarefa Tfcm processar dados transmitidos pelas tarefas Tadc,

Tgpu, Tdcu e Tins, mas perdendo estes mesmos 3×10^{-3} s para processar os dados retornados das tarefas Tace1 e Tace2.

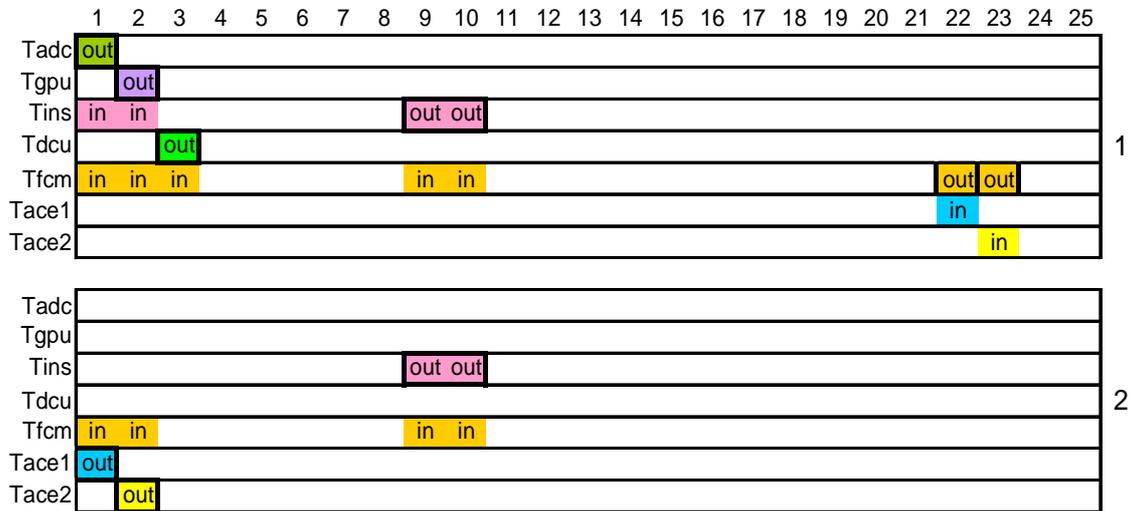


Figura 6.12 – Um ciclo alternativo de produção e consumo de dados.

Um rearranjo mais radical poderia ser o da Figura 6.13, onde as transmissões de dados entre as tarefas Tfcm, Tace1 e Tace2 foram deslocadas para a segunda rodada, ganhando 23×10^{-3} s para a tarefa Tfcm desde a recepção dos dados da tarefa Tins na primeira rodada até o mesmo instante na segunda rodada. A tarefa Tfcm poderia ignorar os dados recebidos da tarefa Tins na segunda rodada ou usá-los para, por exemplo, melhorar uma estimativa com um gasto menor de tempo de processamento.

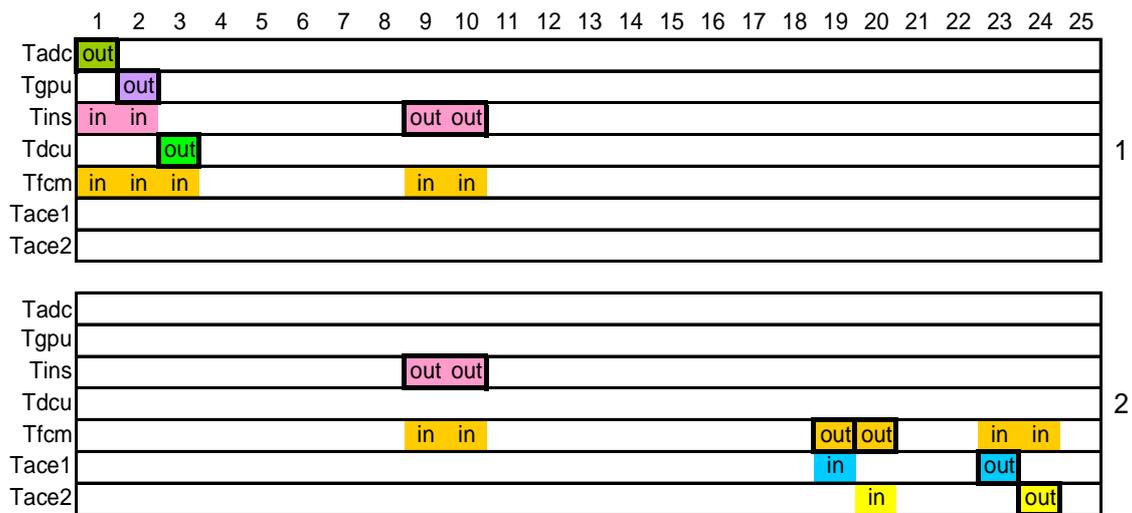


Figura 6.13 – Outro ciclo alternativo de produção e consumo de dados.

Em poucas palavras, o fato de transmissões de dados poderem ser arranjadas no tempo de modo a eliminar conflitos de acesso a um meio físico compartilhado, assegurando total previsibilidade e, portanto, um caráter rigorosamente determinístico na comunicação de dados entre tarefas, não garante por si só a realização da função maior do sistema. É necessário, além disto, averiguar se todas as tarefas, uma vez tendo disponíveis todos os dados necessários, têm tempo suficiente para terminar seu processamento antes de serem novamente agendadas. Com isto, a análise relativamente simples conduzida na Seção 6.2 não poderia ser imediatamente utilizada, pois assume que mensagens seriam transmitidas em sequência sem intervalos.

Uma estratégia que consiga estabelecer uma sequência de transmissões de dados levando em conta os tempos que as tarefas necessitam para terminar seu processamento não é trivial. Em projetos que usam “Time-Triggered Protocol” (TTA-GROUP, 2003), a sequência de transmissão de dados é gerada dissociada do início e do término de processamento das tarefas envolvidas. Em implementações assim, muitas vezes assume-se que uma tarefa pode receber dados em um ciclo, processar e transmitir o resultado no ciclo seguinte. Se isto é adequado, cabe ao projetista do sistema decidir.

Existem ferramentas comerciais que são capazes de análise considerando comunicação e processamento de dados simultaneamente. Uma delas (SymTA/S), desenvolvida pela Syntavision (2008), usa uma técnica baseada em “modelo de eventos” (“event model”) para capturar distorções não-determinísticas em sistemas que operam com eventos periódicos, permitindo analisar a exequibilidade da implementação bem como a determinação de atrasos e suas variações. Uma manifestação acadêmica de emprego de técnica similar (MAST) foi desenvolvida pela Universidad Cantábria (2008). Em geral, estas ferramentas são uma evolução do método introduzido por Tindell e Clark (1994) para obtenção de tempos de resposta em sistemas distribuídos.

A análise baseada em um modelo “Produtor-Consumidor” expõe claramente uma dimensão não examinada na análise da Seção 6.3 anterior, embora não forneça uma solução analítica para a determinação de uma sequência de transmissão de dados que leve em conta os tempos de processamento das tarefas do modelo em bastidor do corretor de trajetória. Contudo, recursos menos sofisticados como, por exemplo, a construção de gráficos como os das Figuras 6.10, 6.11, 6.12 e 6.13, podem auxiliar o projetista a visualizar melhor que ajustes ainda podem ser feitos sem grandes alterações de projeto.

7 CONCLUSÕES E PROPOSTAS PARA ESTUDOS FUTUROS

7.1 Conclusões

No Capítulo 2, foram introduzidos alguns conceitos e conhecimentos de grande interesse atual e de grande utilidade para entendimento das análises feitas nos capítulos posteriores. Entre eles, destaca-se o “Network Calculus”, uma teoria formulada para explicar comportamentos em redes de comunicação de dados, que introduz os conceitos de Curva de Chegada e Curva de Serviço e com eles obtém limites para o atraso de serviço, o acúmulo de tráfego e o fluxo de saída em um sistema que serve um fluxo de dados modelado por uma função cumulativa.

No Capítulo 3, foi avaliado se a abordagem “Rate Monotonic” de Liu e Layland (1973), usada para testar a agendabilidade de um conjunto de tarefas que compartilham um sistema com uma única CPU, poderia ser aplicada substituindo tarefas que executam por mensagens transmitidas em um meio físico compartilhado. Como a transmissão de uma mensagem, ao contrário da execução de uma tarefa, não pode ser interrompida uma vez iniciada, foi concluído que o teste de agendabilidade de Liu e Layland não pode ser imediatamente adaptado porque fere a hipótese de agendamento preemptivo assumida para a aplicação do teste. Partindo desta constatação, uma regra simples foi elaborada para assegurar que todas as mensagens sejam transmitidas com as frequências fixadas.

O Capítulo 4 introduziu o modelo do Corretor de Trajetória usado para as análises dos capítulos posteriores, descrevendo as funções desempenhadas pelas unidades componentes.

O Capítulo 5 apresentou intencionalmente uma implementação dita “federada”, onde cada unidade funcional do sistema opera autonomamente comunicando-se com as outras por meio de ligações físicas ponto-a-ponto. Esta

implementação tem a virtude de evidenciar as interações entre as unidades, facilitando a compreensão das outras implementações nos capítulos seguintes. A título de ilustração, foi avaliada a ocupação de um canal de comunicação baseado no meio físico ARINC 429 entre as duas unidades mais solicitadas (INS e FCM). Em implementações “federadas”, o caráter determinístico na comunicação de dados é assegurado pela capacidade que cada unidade deve apresentar desde seu projeto de receber, processar e transmitir mensagens na frequência e tamanhos especificados sem perdas.

O Capítulo 6 introduziu uma implementação “integrada” do modelo do Corretor de Trajetória baseada em uma topologia de rede em “estrela” tendo como elemento central um “switch” de tecnologia “shared memory”. A porta do “switch” ligada à unidade FCM do Corretor de Trajetória foi modelada usando Curvas de Chegada e Serviço do “Network Calculus”. Em primeiro lugar, todas as mensagens transmitidas à unidade FCM foram consideradas de mesma prioridade e um limite de atraso de serviço igual ao período da mensagem com a maior frequência de transmissão foi usado para determinar a velocidade mínima requerida do meio físico de transmissão e o limite do tráfego retido na porta do “switch” ligado à unidade FCM. Os resultados obtidos com esta análise foram comparados com os resultados obtidos usando a regra simples do Capítulo 3. Em seguida, as mensagens foram classificadas de “baixa” ou “alta” prioridade dependendo da sua frequência (prioridade “alta” para frequências mais altas), a análise foi repetida e os resultados comparados. Nesta implementação “integrada”, o caráter determinístico na comunicação de dados é assegurado pela obtenção analítica de dados de especificação do “switch” (quantidade de memória requerida na porta de saída) e do meio físico da rede de comunicação (velocidade de transmissão de dados) que, se respeitados, devem assegurar transmissões de dados sem perdas.

Ainda no Capítulo 6, uma outra implementação baseada em uma topologia em “barramento” foi apresentada e uma abordagem usando a técnica de “Time Division Multiple Access” (TDMA) foi utilizada para criar um ciclo repetitivo de

transmissões que assegura o caráter determinístico da comunicação de dados entre as unidades do modelo do Corretor de Trajetória e determina a velocidade mínima requerida do meio físico de transmissão. Os resultados desta análise foram comparados com os resultados anteriores obtidos com a implementação usando uma topologia em “estrela” e foi mostrado que o modelamento usando Curvas de Chegada e Serviço do “Network Calculus” produz valores mais conservadores para este último parâmetro de projeto. Como complemento, foi apresentado o algoritmo Welch-Lynch de sincronização de relógios em sistemas distribuídos, recurso essencial para a implementação da comunicação de dados em um meio físico compartilhado usando a estratégia TDMA.

Em seguida, esta análise foi refinada usando um modelo “Produtor-Consumidor”, que leva em conta não só a realização da comunicação de dados, mas também a sequência de interações entre as tarefas que executam nas unidades do Corretor de Trajetória para o desempenho de sua função maior. Com esta análise, foi mostrado que é insuficiente assegurar o caráter determinístico das transmissões de dados em um sistema distribuído que opera em rede, porque é fundamental assegurar também que todas as tarefas envolvidas, que precisam responder a recepções e iniciar transmissões de dados, têm tempo suficiente para terminar o seu processamento. Sem esta composição, o Corretor de Trajetória pode deixar de cumprir seus requisitos de projeto mais primários e falhar catastroficamente na sua vida em operação.

7.2 Propostas para Possíveis Trabalhos Futuros

As análises desenvolvidas neste trabalho são suficientes para assegurar a realização física de uma parte importante do projeto de uma unidade embarcada responsável total ou parcialmente pelo sucesso de uma missão: uma comunicação de dados com características determinísticas, que opera sem perdas por um meio físico de transmissão implementado usando alguma topologia de rede. Com as abordagens apresentadas, é possível conhecer,

ainda em fase de projeto, limites para atrasos de transmissão, consumo de memória nas portas de entrada e saída e velocidades mínimas necessárias ao canal de comunicação, permitindo a escolha correta de um meio físico e de uma estratégia apropriada para a transmissão de dados.

Contudo, há evidências de outras abordagens que conseguem obter tempos de resposta levando em conta toda uma cadeia de eventos, periódicos ou não, que causam a ativação, início e término de execução de tarefas em um sistema distribuído que opera usando um canal de comunicação compartilhado. Uma continuação deste trabalho poderia investigar mais profundamente estas abordagens, avaliar os seus limites de aplicação e seus resultados, preferencialmente usando o mesmo modelo do Corretor de Trajetória para permitir comparações entre as novas abordagens e as aqui apresentadas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALENA, R. L.; OSSENFORT, J. P.; LAWS, K. I.; GOFORTH, A.; FIGUEROA, F. Communications for Integrated Modular Avionics. In: IEEE AEROSPACE CONFERENCE (IEEEAC), 2006, Big Sky, Montana. **Proceedings...** Montana: IEEE, 2006. Paper #1230.

ANSWERS CORPORATION. **Free on-line dictionary**. Disponível em: www.answers.com. Acesso em: 28-Nov-2008.

AERONAUTICAL RADIO INCORPORATED. **ARINC specification 429**: Mark 33 Digital Information Transfer System (DITS) – Part 1 Functional Description, Electrical Interface, Label Assignments and Word Formats, 2001.

AERONAUTICAL RADIO INCORPORATED. **ARINC specification 653-1**: Avionics Application Software Standard Interface, Out. 2003.

AERONAUTICAL RADIO INCORPORATED. **ARINC specification 664**: Aircraft Data Networks – Part 7, Deterministic Networks, 2003.

BURNS, A.; WELLINGS, A. **Real-time systems and programming languages**, 2. ed. New York, NY, USA: Addison-Wesley, 1997.

CONTROLLER AREA NETWORK (CAN). **CAN specification version 2.0**. Stuttgart, Alemanha: Robert Bosch, Set. 1991.

CRUZ, R. L. A Calculus for Network Delay: Parts I and II. **IEEE Transactions on Information Theory**, v. 37, n. 1, Jan. 1991.

DUTERTRE, B. **The Welch-Lynch clock synchronization algorithm**. London, UK. University of London. Department of Computer Science Queen Mary and Westfield Colleg, 1998. Technical Report 747.

ENTERASYS SECURE NETWORKS. **Matrix N-series**. Disponível em: www.enterasys.com. Acesso em: 28-Nov-2008.

GEORGES, J.-P.; DIVOUX, T.; RONDEAU, E. Confronting The Performances Of A Switched Ethernet Network With Industrial Constraints By Using The Network Calculus. **International Journal of Communication Systems**, v. 18, p. 877-903, 2005.

GRIEU, J. **Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques**. Tese para obtenção do grau de Doutor en Réseaux et Télécommunications no Institut National Polytechnique de Toulouse, Toulouse, França, 2004.

INSTITUTE OF ELECTRICAL AND ELECTRONIS ENGINEERS (IEEE). **IEEE standard 802.3**: LAN/MAN CSMA/CD Access Method. Piscataway, NJ, USA, 2005.

IP/MPLS FORUM. **ATM forum specifications**. Disponível em: www.ipmplsforum.org. Acesso em: 28-Nov-2008.

LAROUSSE CULTURAL. **Dicionário da língua portuguesa**. São Paulo, SP, Brasil: Editora Nova Cultural, 1992.

LAZOWSKA, E.; ZAHORJAN, J.; GRAHAM, G.; SEVCIK, K. **Quantitative system performance** – computer system analysis using queueing network models. Englewood Cliffs, NJ, USA: Prentice Hall, 1984.

LE BOUDEC, J. Y.; THIRAN, P, **Network Calculus** - a theory of deterministic queueing systems for the Internet. Berlin, Alemanha: Springer-Verlag, 2004.

LIU, J. W. S. **Real-time systems**. 1. ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Jun. 2000.

LIU, C. L.; LAYLAND, J. W. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. **Journal of the Association for Computing Machinery**, v. 20, n. 1, Jan. 1973.

MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 3. ed. Rio de Janeiro, RJ, Brasil: Editora LTC, 2002.

MARTIN, J.; CHAPMAN, K. K. **Local Area Networks** – architectures and Implementations. Englewood Cliffs, NJ, USA: Prentice Hall, 1989.

MILITARY STANDARD. **MIL-STD-1553B**: Aircraft Internal Time Division Command/Response Multiplex Data Bus. United States Department of Defense, Set. 1987.

MORAIS, C. G. **Modelagem e desempenho de servidores web Empregando a teoria Network Calculus**. Dissertação para obtenção do grau de Mestre em Informática na Universidade Federal do Paraná, UFPR, Curitiba, PR, Brasil, 2005.

RADIO TECHNICAL COMMISSION FOR AERONAUTICS (RTCA). **RTCA DO-297**: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations. Washington D.C., USA, Nov. 2005.

SANZ, R.; ÁRZÉN, K. Trends in Software and Control. **IEEE Control Systems Magazine**, v. 23, n. 3, 2003

SEIFERT, R. **The switch book** – the complete guide to LAN switching technology. New York, NY, USA: John Wiley & Sons, 2000.

STANKOVIC, J. A. Misconceptions about Real-Time Computing: A Serious Problem for Next Generation Systems. **IEEE Computer**, v. 21, p. 10-19, Out. 1988.

STILIADIS, D.; VARMA, A. Latency-Rate Servers: A General Model for Traffic Scheduling Algorithms. **IEEE/ACM Transactions on Networking**, v. 6, n. 5, p. 611-624, USA, Out. 1998.

SUN, J.; LIU, J. **Bounding the end-to-end response times of tasks in a distributed real-time system using the direct synchronization protocol**, 1996. Disponível em <http://pertserver.cs.uiuc.edu/papers/.SuLi96a.ps>. Acesso em 28-Nov-2008.

SYMTAVISION. **SymTAS scheduling analysis tool suite**. Disponível em: www.symtvision.com. Acesso em: 28-Nov-2008.

TINDELL, K.; CLARK, J. Holistic schedulability analysis for distributed hard real-time systems. **Microprocessing & Microprogramming**, v. 50, n. 2-3, p. 117-134, Abr. 1994.

TTA-GROUP. **Time-Triggered Protocol (TTP/C) high-level specification document** – protocol version 1.1, Viena, Áustria: TTA-GROUP, 2003.

UNIVERSIDAD CANTÁBRIA. **MAST – Modelling and analysis suite for real-time applications**. Disponível em: <http://mast.unican.es/#intro>. Acesso em: 28-Nov-2008.

VICTORIA PROJECT. **Validation platform for integration of standardised components, technologies, and tools in an open, modular and improved aircraft electronic system**. 5th Framework Programme of the European Commission, 2001-2004. Disponível em: http://www-v3.thalesgroup.com/all/pdf/cp_victoria.pdf. Acesso em: 28-Nov-2008.

WELCH, J. L.; LYNCH N. A New Fault-Tolerant Algorithm for Clock Synchronization. **Information and Computation**, v. 77, p.1-36, Abr. 1988.

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.