

31453-0

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TEMPOS DE COMUNICAÇÃO EM
MULTIPROCESSADORES

por

JOÃO PAULO FUMIO WHITAKER KITAJIMA

Dissertação submetida como requisito parcial para
a obtenção do grau de mestre em
Ciência da Computação



SABi



Prof. Philippe D. A. Navaux
Orientador

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Porto Alegre, novembro de 1990

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Nº CHAMADA 681.32.02(043) K62T		º REG: 6228
		DATA: 28/11/90
ORIGEM D	AT.: 28/11/90	P. CO: 4000,00
FUNDO: II	FORN.: CPGCC	

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Biblioteca da UFRGS

6228

KITAJIMA, JOAO PAULO FUMIO
WHITAKER

TEMPOS DE COMUNICACAO EM
MULTIPROCESSADORES

681.32.02(043)
K62T

INF
1991/31453-0
1990/11/28

MOD. 2.3.2

"Tudo vale a pena
Se a alma não é pequena"
(Fernando Pessoa)

Page 1 of 1

The following information is provided for your reference:

1. The first section of the document discusses the importance of maintaining accurate records.

2. The second section details the procedures for handling confidential information.

3. The third section outlines the responsibilities of all personnel involved in the process.

4. The fourth section provides a summary of the key findings and recommendations.

5. The fifth section contains the conclusions and next steps for the project.

AGRADECIMENTOS

Ao Navaux, pela orientação, pela ajuda e pelo empenho demonstrado em relação ao meu estágio em Grenoble, onde foi iniciada esta dissertação. Ao Tazza, também pela orientação durante parte do mestrado e pelo interesse. À Brigitte Plateau que, com sua dedicação, confiança e paciência, sugeriu-me o tema deste trabalho e muito contribuiu na fase inicial de pesquisa (novembro de 1989). Aos alunos, auxiliares de pesquisa, professores e funcionários do CPGCC. À CAPES e ao CNPq, pelo apoio financeiro destes quase trinta meses de labuta. Aos meus pais e irmão, sem os quais nada seria possível. À Carmem, Maria Cláudia e Siley, pelo apoio aí no Planalto Central.

SUMÁRIO

GLOSSÁRIO	9
LISTA DE FIGURAS	13
RESUMO	15
ABSTRACT	17
1 INTRODUÇÃO	19
2 AVALIAÇÃO DE DESEMPENHO DE MÁQUINAS PARALELAS ...	25
3 MODELAGEM DE SISTEMAS	33
4 COMUNICAÇÃO EM MULTIPROCESSADORES	37
4.1 <u>Escopo da modelagem</u>	37
4.2 <u>Comunicação</u>	38
4.3 <u>Estratégias de comunicação</u>	43
4.3.1 A estratégia comutação de mensagens ...	44
4.3.2 A estratégia "virtual cut-through"	45
4.3.3 A estratégia "rendez-vous"	46
4.3.4 A estratégia "wormhole"	47
5 MODELOS ANALÍTICOS	51
5.1 <u>Modelo analítico : comutação de mensagens</u> ...	53
5.2 <u>Modelo analítico : "virtual cut-through"</u>	55
5.3 <u>Modelo analítico : "rendez-vous"</u>	56
5.4 <u>Modelo analítico : "wormhole"</u>	59
6 MODELOS DE SIMULAÇÃO	63
6.1 <u>Modelo de simulação : "rendez-vous"</u>	64
6.2 <u>Modelo de simulação : "wormhole"</u>	71
7 ANÁLISE DOS MODELOS	75
7.1 <u>Implementação dos modelos</u>	75
7.2 <u>Validação</u>	76
7.3 <u>Análise das estratégias</u>	81
7.4 <u>Estudo de caso</u>	90

8 CONCLUSÃO	95
ANEXO 1 - TEORIA DE FILAS - O MODELO M/M/1 [KLE 75]	101
ANEXO 2 - GPSS [JAC 80]	111
ANEXO 3 - PROGRAMAS EM GPSS DO MODELO DE SIMULAÇÃO DO "RENDEZ-VOUS"	117
ANEXO 4 - PROGRAMAS EM GPSS DO MODELO DE SIMULAÇÃO DO "WORMHOLE"	127
ANEXO 5 - RESULTADOS NUMÉRICOS DAS SIMULAÇÕES EFETUADAS	139
BIBLIOGRAFIA	153

GLOSSÁRIO

ACK

Acrônimo : ACKnowledgement - reconhecimento, agradecimento.

BENCHMARK

Tradução (expressão da língua inglesa) : "marca de nível". Técnica de avaliação de desempenho de computadores. Consiste em medir o tempo necessário para um computador executar uma série de programas padrão [KOB 78].

BROADCAST

Tradução (expressão da língua inglesa) : "difusão". Emissão de uma mensagem de um processador emissor para vários processadores destinos [CHA 84].

CACHE

Tradução (expressão da língua inglesa) : "esconderijo". Memória de acesso rápido. Contém dados e instruções mais frequentemente referenciados pelo processador [HWA 84].

CUT-THROUGH

Tradução (expressão da língua inglesa) : "cortar ou atravessar algo ligeiramente". Estratégia de comunicação entre dois processadores. Características : não reserva canais e necessita hardware especial [KER 79].

DEADLOCK

Tradução (expressão da língua inglesa) : "bloqueio mortal". Fenômeno de bloqueio mútuo entre processos [DAL 87].

FEEDBACK

Tradução (expressão da língua inglesa) : "alimentar de volta". Mecanismo de retro-alimentação (teoria de sistemas).

FLITS

Acrônimo : "FLow control digITS" - dígitos de controle de fluxo [DAL 87].

GPSS

Abreviatura : General Purpose Simulator System - Sistema Simulador de Propósito Geral [DON 79].

HOLDING TIME

Tradução (expressão da língua inglesa) : "tempo de alocação". Fração do tempo correspondente à alocação de um canal por uma dada comunicação.

HOPS

Tradução (expressão da língua inglesa) : "salto". Um canal de comunicação. Expressão utilizada para avaliação da distância percorrida por uma mensagem em uma rede de processadores [KER 79].

LOOSELY COUPLED

Tradução (expressão derivada da língua inglesa) : "fracamente acoplado". Tipo de multiprocessador [HWA 84].

MESSAGE SWITCHING

Tradução (expressão da língua inglesa) : "comutação de mensagem". Estratégia de comunicação entre dois processadores. Características : não reserva canais e não necessita de hardware especial [KER 79].

RENDEZ-VOUS

Tradução (expressão da língua francesa) : "encontro, reunião". Estratégia de comunicação entre dois processadores. Inspirada no protocolo físico de comunicação implementado em "transputers". Características : reserva canais e não necessita de hardware especial [PLA 89].

SWITCH

Tradução (expressão da língua inglesa) : "chave de comutação". Dispositivo de comutação de mensagens.

TIGHTLY COUPLED

Tradução (expressão derivada da língua inglesa) : "fortemente acoplado". Tipo de multiprocessador [HWA 84].

TRANSPUTER

Marca registrada da empresa inglesa INMOS. Microprocessador [JES 88][TRY 89].

WORMHOLE

Tradução (expressão da língua inglesa) : "buraco de verme". Estratégia de comunicação entre dois processadores. Características : segmentação da mensagem, hardware especial e reserva de canais [DAL 87].

... ..
... ..
... ..

... ..

... ..
... ..
... ..

... ..
... ..
... ..
... ..

... ..

... ..
... ..
... ..
... ..

LISTA DE FIGURAS

Figura 2.1 Trabalho em grupo [FOX 87]	26
Figura 2.2 Multiprocessadores de fraco acoplamento	28
Figura 2.3 Multiprocessadores de forte acoplamento	28
Figura 2.4 Técnicas de avaliação de desempenho	30
Figura 3.1 Etapas do processo de modelagem auxiliada por computador [JAC 80]	33
Figura 4.1 Âmbito do capítulo 4	37
Figura 4.2 Formato de uma mensagem	39
Figura 4.3 Bloqueio mortal [DAL 87]	40
Figura 4.4 Tempo de comunicação	41
Figura 4.5 Modelo preliminar de comunicação	43
Figura 4.6 A estratégia comutação de mensagens	45
Figura 4.7 A estratégia "virtual cut-through"	46
Figura 4.8 A estratégia "rendez-vous"	48
Figura 4.9 A estratégia "wormhole"	49
Figura 5.1 Âmbito do capítulo 5	51
Figura 5.2 Modelagem da comutação de mensagens	54
Figura 5.3 Tempo médio de alocação do canal no "rendez-vous"	58
Figura 5.4 Tempo médio de alocação do canal no "wormhole"	60
Figura 6.1 Âmbito do capítulo 6	63
Figura 6.2 Fluxograma do modelo de simulação do "rendez-vous"	65
Figura 6.3 Seqüência linear de processadores	67
Figura 6.4 Anel de processadores	67
Figura 6.5 Fluxograma do modelo de simulação do "wormhole"	72
Figura 7.1 Âmbito do capítulo 7	75
Figura 7.2 Resultado da comparação entre topologias	78
Figura 7.3 Validação do "cut-through"	80
Figura 7.4 Validação (modelo analítico x modelo em GPSS)	82
Figura 7.5 Variação dos parâmetros de comunicação .	89
Figura 7.6 Topologia reconfigurável	91

Figura 7.7 Topologia estática	91
Figura 8.1 Síntese das estratégias de comunicação .	97
Figura A.1 Representação gráfica de uma fila	102
Figura A.2 Diagrama de transições de estados para M/M/1	104

RESUMO

Na pesquisa por novas maneiras de se obter maior poder de processamento dos computadores, o paralelismo é considerado uma alternativa viável. Mas a replicação de processadores não representa por si só um avanço nestas pesquisas. Problemas surgiram, antes inexistentes no paradigma seqüencial: paralelização da solução, mapeamento na arquitetura alvo, balanceamento da carga da máquina paralela, comunicação e sincronização, entre outros.

Em particular, a comunicação entre processos em um multiprocessador fracamente acoplado é um aspecto crucial que afeta o desempenho deste tipo de sistema como um todo. Quatro estratégias de comunicação entre processadores são apreciadas neste trabalho: comutação de mensagens ("message switching"), "virtual cut-through", "rendez-vous" e "wormhole". Para cada caso, modelos analíticos (baseados em teoria de filas) e de simulação discreta são desenvolvidos e aplicados a fim de determinar, dentro de certos contextos, qual a melhor estratégia. O "cut-through" e a comutação de mensagens (esta última não depende de hardware específico) são as melhores políticas para sistemas com elevado grau de comunicação (os modelos destas estratégias, utilizados neste trabalho, já foram desenvolvidos na literatura por Kermani e Kleinrock). O "wormhole", que apresenta características de reserva, pode ser apropriado para sistemas com pouca troca de mensagens. "Rendez-vous" não depende de hardware especial, mas apresenta maior tempo de comunicação em relação às outras estratégias.

Os modelos descritos foram construídos de acordo com uma metodologia passo-a-passo e modular. Esta metodologia é também apresentada e fundamenta a linha de raciocínio desenvolvida durante a apresentação dos diferentes capítulos desta dissertação.

Page 1

Faint, illegible text covering the majority of the page, likely bleed-through from the reverse side.

ABSTRACT

In the research for more computer processing power, parallelism is a feasible alternative. But the processor replication alone doesn't represent an advance in this field. New problems, absent in the sequential paradigm, have appeared: solution parallelization, mapping, load balancing, synchronization, communication and others.

The communication between processes in loosely-coupled multiprocessors affects the system performance as a whole. Four interprocessor communication strategies are analyzed in this work: message switching, virtual cut-through, "rendez-vous" and wormhole. For each case, analytic (based on queueing theory) and simulation models are developed and applied in order to determine which strategy is the best and under which contexts. Cut-through and message switching (this last strategy doesn't depend on specific hardware) are better for heavy-loaded systems (these strategies were already modelled by Kermani and Kleinrock). Wormhole (presenting blocking and reserving aspects) can be more suitable for systems with low communication level. "Rendez-vous" doesn't depend on special hardware, but generates longer communication times than those generated by the other communication strategies.

The models described were developed according to a step-by-step and modular methodology. This method is also presented and gives logical support to the work through the different chapters.

The first part of the report deals with the general situation of the country and the progress of the war. It is followed by a detailed account of the military operations in the West, and then a chapter on the situation in the East. The report concludes with a summary of the main points and a list of references.

The second part of the report is devoted to a detailed analysis of the military operations in the West. It begins with a description of the German offensive in the Ardennes region, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the Low Countries and the progress of the war in the East.

The third part of the report is a detailed account of the military operations in the East. It begins with a description of the German offensive in the East, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the Balkans and the progress of the war in the Pacific.

The fourth part of the report is a detailed account of the military operations in the Pacific. It begins with a description of the Japanese offensive in the Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the Philippines and the progress of the war in the South Pacific.

The fifth part of the report is a detailed account of the military operations in the South Pacific. It begins with a description of the Japanese offensive in the South Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the New Guinea and the progress of the war in the South West Pacific.

The sixth part of the report is a detailed account of the military operations in the South West Pacific. It begins with a description of the Japanese offensive in the South West Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the New Guinea and the progress of the war in the South West Pacific.

The seventh part of the report is a detailed account of the military operations in the South West Pacific. It begins with a description of the Japanese offensive in the South West Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the New Guinea and the progress of the war in the South West Pacific.

The eighth part of the report is a detailed account of the military operations in the South West Pacific. It begins with a description of the Japanese offensive in the South West Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the New Guinea and the progress of the war in the South West Pacific.

The ninth part of the report is a detailed account of the military operations in the South West Pacific. It begins with a description of the Japanese offensive in the South West Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the New Guinea and the progress of the war in the South West Pacific.

The tenth part of the report is a detailed account of the military operations in the South West Pacific. It begins with a description of the Japanese offensive in the South West Pacific, and then goes on to discuss the Allied counter-offensive. The report also covers the situation in the New Guinea and the progress of the war in the South West Pacific.

The report concludes with a summary of the main points and a list of references.

1 INTRODUÇÃO

O processamento paralelo é definitivamente uma solução viável para atender à elevada demanda por alto desempenho. A máquina monoprocessadora conseguiu resolver apenas um número limitado, ainda que considerável, de problemas. À medida que os sistemas manipulados pelo homem tornam-se mais e mais complexos, os problemas computacionais acompanham esta tendência e novas arquiteturas de máquinas e novas metodologias de programação tornam-se centros de interesse por parte dos pesquisadores vinculados à Ciência da Computação.

A filosofia do paralelismo é a do "trabalho em grupo": um problema é resolvido muito mais rápido por um conjunto de elementos do que por um elemento sozinho. Pode-se comparar esta situação com a construção de uma casa, onde vários operários trabalham em conjunto a fim de terminar mais rapidamente a obra.

Mas o paralelismo não é a solução para todos os problemas. Ao contrário, ele traz consigo uma série de novas outras questões que antes não existiam nas máquinas seqüenciais. Estes problemas podem comprometer os benefícios da paralelização de algoritmos e devem ser adequadamente tratados.

Alguns destes problemas podem ser compreendidos, se a analogia com a construção de uma casa for utilizada. A questão primeira é a distribuição de tarefas ou como paralelizar a solução. Outro aspecto consiste no mapeamento do algoritmo na arquitetura alvo: os módulos paralelos são distribuídos entre os processadores do computador. Na construção, corresponde à alocação das tarefas aos operários e qual a ordem de execução das mesmas. No decorrer da execução do algoritmo, alguns processadores podem apresentar um grau maior de utilização do que outros. É necessário,

desta forma, balancear a carga de trabalho entre os diversos recursos disponíveis. Também, na construção de uma casa, os operários precisam, de alguma forma, trocar informações e avisar ao chefe da obra qual a situação de cada tarefa (parada, em andamento, concluída). A comunicação é importante na resolução de um algoritmo paralelo e, na verdade, consiste em um fator crucial no desempenho da aplicação. A comunicação é fundamental pois afeta o algoritmo não só durante a sua execução, mas exerce influência também durante todo o seu projeto. O balanceamento da carga, a sincronização dos diversos processos concorrentes e a definição de um mapeamento adequado dependem de um sistema eficiente de comunicação.

Para dois processadores distintos se comunicarem, diferentes protocolos sobre diferentes configurações podem ser idealizados. Um certo conhecimento a respeito da comunicação em arquiteturas paralelas foi absorvido da experiência adquirida com a construção e avaliação de redes de computadores, tais como a ARPANET [KLE 76]. Porém, características físicas entre os dois tipos de redes (inter e intracomputador) fazem com que as abordagens sejam diferentes, fato que reflete também na camada de software que está sobre estes sistemas. Por exemplo, as ligações entre os processadores de uma máquina paralela são mais confiáveis do que as de uma rede de computadores. Os protocolos de comunicação podem ser desta forma mais simples e mais eficientes.

Quatro estratégias são analisadas: comutação de mensagens ("message switching"), "virtual cut-through", "rendez-vous" e "wormhole". A comutação de mensagens se assemelha à comutação de pacotes em redes de computadores, onde as mensagens contêm um endereço de destino. Estas mensagens podem passar por processadores distintos até chegarem ao seu destino final. Em cada processador, o endereço é analisado e, de acordo com regras de roteamento

(determinação de caminhos para comunicação), a mensagem é transmitida pela linha de saída adequada. Se esta linha de saída estiver ocupada, a mensagem é armazenada em uma fila de espera. O "virtual cut-through" possui comportamento semelhante ao da comutação de mensagens. Porém, se o canal de saída estiver desocupado, não há retransmissão da mensagem: ela passa pelo processador sem provocar qualquer atraso de transmissão. Neste caso, a análise do endereço da mensagem é feita em tempo real por um dispositivo físico. Tal dispositivo é capaz de passar diretamente a mensagem do canal de entrada para o canal de saída, sem acarretar processamento no nó. O enfileiramento da mensagem neste nó somente ocorre se o canal de saída estiver ocupado. O "rendez-vous" se assemelha a um protocolo de rede, onde um processador envia uma solicitação de comunicação, recebe uma aceitação e, finalmente, envia a mensagem por uma seqüência de caminhos já reservados. No "wormhole" também há reserva de caminhos, porém a mensagem é particionada em pequenos pedaços (denominados "flits") e somente um grupo de "flits" é transmitido de cada vez.

A comutação de mensagens e o "wormhole" foram modelados na literatura [KER 79] e as demais estratégias, neste trabalho. Os modelos são validados contra resultados de simulação. Aos modelos válidos são aplicados parâmetros para determinação da melhor estratégia. Todavia, a "melhor estratégia" depende da combinação de uma série de fatores. Desta forma, outro objetivo é especificar quais destes fatores são preponderantes e qual a relação entre os mesmos que torna esta ou aquela estratégia mais apropriada.

Os modelos desenvolvidos consideram um número reduzido de parâmetros e hipóteses "simplistas", na maioria das vezes incoerentes com a realidade. Mas existem dois motivos para utilizar os modelos:

- validade dos modelos analíticos e das aproximações. Trabalhos encontrados na literatura comprovam que mesmo modelos construídos em cima de premissas simplificadoras apresentaram resultados coerentes com a realidade [SCH 67]
- os resultados aqui apresentados têm caráter de escolha de alternativas, ou seja, deseja-se saber qual a melhor estratégia e não uma quantificação exata desta superioridade. Os modelos podem ser utilizados em uma etapa inicial do projeto de arquiteturas.

A modelagem das estratégias bloqueantes ("rendez-vous" e "wormhole") apresentou dificuldades, contornadas através de aproximações. Os modelos são pessimistas. A discrepância tornou-se mais evidente em situações de alta utilização de recursos e de maior distância entre os dois processadores comunicantes. Mesmo com resultados pessimistas, os modelos foram utilizados para uma primeira comparação entre as diversas estratégias. Os mecanismos onde o paralelismo da comunicação era pouco explorado (comutação de mensagens e, principalmente, o "rendez-vous") apresentaram o pior desempenho. Sob baixa utilização de canais, o "wormhole" é a estratégia mais adequada. Para ambientes com moderadas a altas taxas de comunicação, o "cut-through" e a comutação de mensagens se apresentaram mais eficientes.

O trabalho foi dividido em 8 capítulos e 3 anexos. O próximo capítulo apresenta com mais detalhes o processamento paralelo, seus benefícios e obstáculos, e métodos de avaliação de desempenho de arquiteturas paralelas. O capítulo 3 introduz uma metodologia, empregada neste trabalho, para modelagem de sistemas. O capítulo 4 trata especificamente do problema da comunicação. Nesta etapa, são introduzidas as 4 estratégias citadas

anteriormente e, no capítulo 5, os respectivos modelos. À simulação foi dedicado todo o capítulo seguinte. O capítulo 7 discute a validação e os resultados obtidos através da utilização dos modelos. Cada parâmetro foi analisado independentemente e o estudo de caso tenta, através de uma situação real, analisar a importância destes parâmetros na análise da reconfigurabilidade de multiprocessadores. Finalmente, conclusões são apresentadas e direções de pesquisa discutidas. O primeiro anexo apresenta noções básicas de teoria de filas, principalmente o procedimento para obtenção da expressão do tempo de espera em um sistema de filas M/M/1 (tempo entre chegada de clientes exponencial, tempo de serviço do cliente exponencial e apenas um servidor). O segundo anexo trata de GPSS, a linguagem de simulação empregada neste trabalho. O último anexo exhibe os números utilizados para a construção dos gráficos do capítulo 7.

A idéia de comparar estratégias de comunicação surgiu em decorrência da fabricação de dois tipos de máquinas paralelas : o Meganode [JES 88] que se compõe de uma rede reconfigurável de processadores e a máquina de Seitz [SEI 89], uma rede estática de nós processadores que utiliza o mecanismo "cut-through" para comunicação entre processos. Os dois equipamentos, a grosso modo, ocupam a mesma faixa de mercado. Seitz [SEI 89] afirma que a sua solução apresenta melhor desempenho, mas nenhum estudo comparativo foi apresentado para justificar tal afirmação.

Outro objetivo a ser alcançado é a exercitação de técnicas de modelagem orientadas para análise de desempenho de sistemas computacionais. Aqui, a modelagem analítica e a simulação foram empregadas. A primeira não se mostrou adequada para algumas situações (reserva de recursos). Para estes caso, a simulação, mesmo com suas desvantagens (por exemplo, maior tempo gasto para obtenção dos índices de desempenho), conseguiu suprir a deficiência dos modelos

analíticos.

Por fim, uma observação adicional deve ser feita : certos termos não foram traduzidos, mesmo o nome das estratégias ("wormhole", "rendez-vous", "cut-through"), pelo fato de pertencerem a uma terminologia consagrada na comunidade científica de Informática. Somente a estratégia "message switching" foi traduzida para "comutação de mensagens". A partir deste ponto, as expressões estrangeiras não comuns presentes no glossário estão marcadas com (*) na sua primeira aparição.

2 AVALIAÇÃO DE DESEMPENHO DE MÁQUINAS PARALELAS

Os problemas de caráter computacional exigem, cada vez mais, um maior poder de processamento. Estes problemas pertencem às diversas áreas do conhecimento (engenharia, física, química, economia, medicina) e cobrem uma ampla gama de aplicações (cálculos científicos, simulações, visualizações, inferências).

A demanda por potência de computação pode ser satisfeita : (a) através da estruturação adequada do algoritmo associado à solução do problema ou (b) através de um computador mais eficiente. Um aspecto não está dissociado do outro e, de fato, somente através da combinação das duas alternativas acima, é possível conceber um aumento significativo do desempenho das aplicações.

Os computadores de primeira geração (1938 a 1953 aproximadamente) caracterizavam-se por uma unidade de processamento. Os programas eram executados seqüencialmente, ou seja, uma instrução após a outra. Para conseguir um maior desempenho, além de maior confiabilidade nas operações, os programas eram otimizados, embora permanecessem seqüenciais, e as máquinas eram construídas a partir de componentes mais sofisticados, embora vigorasse ainda o modelo uniprocessador. As gerações seguintes, baseadas nesta filosofia, constituíram a linha convencional de projeto de arquiteturas.

Mesmo que a tecnologia de materiais avançasse a passos largos, projetistas e usuários de aplicações de grande porte não estavam satisfeitos com o computador, que possuía um processador veloz e que executava programas seqüenciais otimizados. Naturalmente, a seqüencialidade dos algoritmos imposta pela seqüencialidade do processador era uma restrição. Portanto, um novo e mais eficiente modelo de processamento deveria ser encontrado.

Este novo modelo surgiu a partir do questionamento sobre a possibilidade de um único algoritmo (solução) ser executado por mais de um processador, assim como uma casa é construída por uma equipe de operários [FOX 87] (figura 2.1). A replicação de processadores implica em mudanças fundamentais no projeto de novas máquinas e no desenvolvimento dos algoritmos, pois traz consigo um aspecto não existente no modelo seqüencial : o processamento paralelo de tarefas.

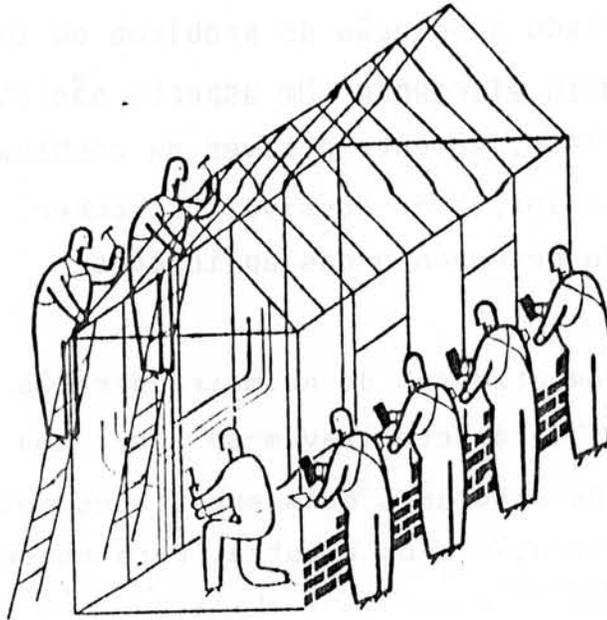


Figura 2.1 - Trabalho em grupo [FOX 87]

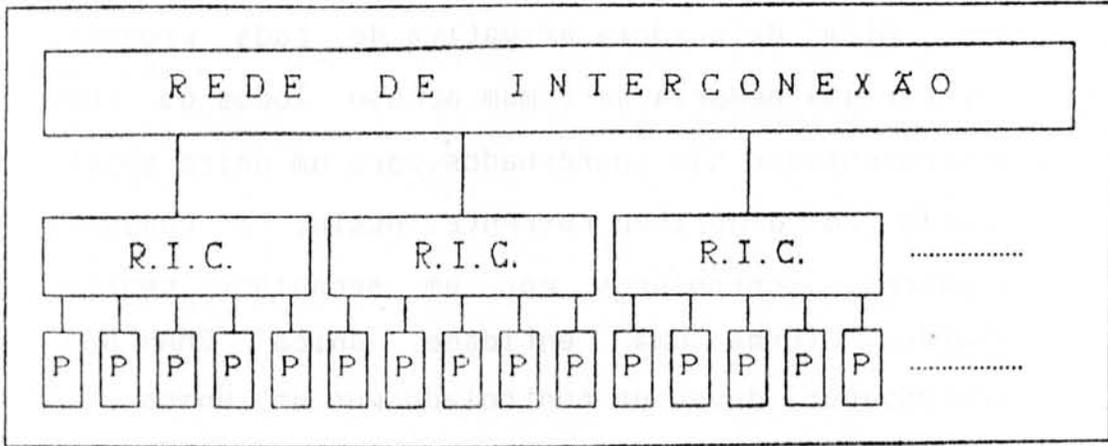
Processamento paralelo é um termo que caracteriza uma gama variada de sistemas, desde redes de computadores até componentes de circuitos integrados. Mesmo arquiteturas uniprocessadoras apresentam certas formas de paralelismo interno. Um exemplo é a existência de dispositivos específicos, denominados canais, que realizam operações de entrada/saída ao mesmo tempo que liberam o processador central para execução de outros programas.

Dentre as diversas arquiteturas paralelas, os multiprocessadores apresentam especial interesse.

Caracterizam-se pela existência de múltiplas unidades de processamento, cada uma capaz de executar seu próprio programa. Além da memória privativa de cada processador, pode existir uma memória de comum acesso. Todos os recursos do multiprocessador são coordenados para um único objetivo : a execução do algoritmo corrente. Assim, o conjunto de processadores, controlados por um mecanismo central ou distribuído, forma uma entidade única [MAR 86]. O multiprocessador deve ser controlado por um único sistema operacional com capacidade de prover, em diversos níveis, as interações entre processadores e programas [HWA 84].

Os multiprocessadores podem ser classificados [HWA 84] em fracamente acoplados ("loosely coupled"(*)) e fortemente acoplados ("tightly coupled"(*)). Os multiprocessadores de fraco acoplamento possuem processadores com interface para dispositivos de entrada/saída e uma memória local capaz de armazenar instruções e dados. Não há uma memória central. Os processadores estão interconectados por canais de comunicação, os quais formam uma rede de interconexão. Programas em diferentes processadores comunicam-se através de mensagens que circulam pela rede de interconexão. As unidades de processamento podem ser agrupadas em conjuntos. Conjuntos podem ser agrupados em outros conjuntos : a estrutura do multiprocessador torna-se hierárquica (figura 2.2).

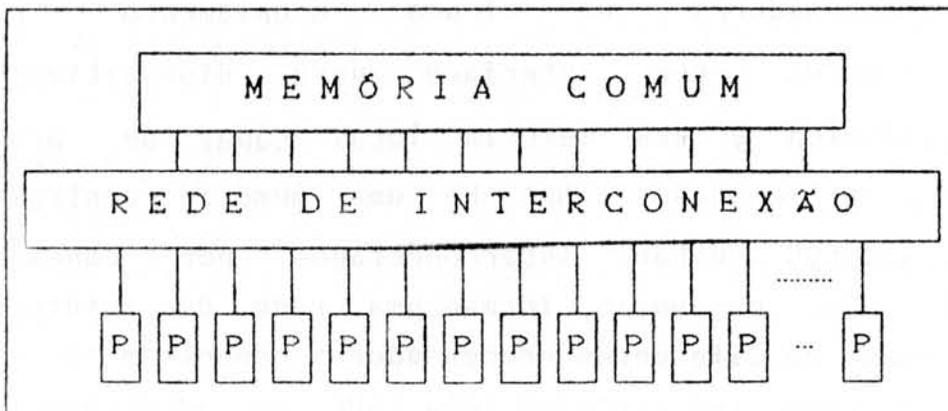
Por outro lado, multiprocessadores fortemente acoplados possuem uma memória de acesso comum a todos os processadores. O sistema de comunicação conecta os diversos processadores à memória comum e qualquer troca de dados é feita através desta memória. Os processadores possuem também uma memória privativa para armazenamento de dados e de instruções frequentemente utilizados (memória "cache"(*) - figura 2.3).



R.I.C. : rede de interconexão do conjunto

P : processador

Figura 2.2 - Multiprocessadores de fraco acoplamento



P : processador

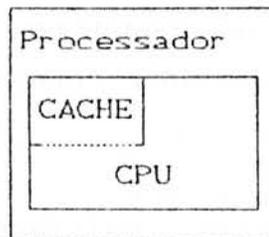


Figura 2.3 - Multiprocessadores de forte acoplamento

Os multiprocessadores representam uma evolução tecnológica em termos de arquitetura de computadores. Evidentemente, o advento do paralelismo e, conseqüentemente, dos multiprocessadores trouxe uma série de novos problemas,

antes inexistentes no paradigma seqüencial. Como exemplos, pode-se citar a paralelização do próprio algoritmo que implementa a solução do problema em questão, o mapeamento [BOK 81] dos módulos paralelos deste algoritmo em um conjunto arbitrário de processadores, o balanceamento da carga [EAG 86] do multiprocessador a fim de se obter uma utilização homogênea do mesmo e a determinação de um esquema adequado de suporte à comunicação entre os diversos processos paralelos.

A avaliação do impacto destes problemas e a análise de soluções tornam-se imprescindíveis, a fim de validar os benefícios advindos com o processamento paralelo. Três aspectos devem ser considerados no estudo do desempenho de máquinas paralelas : metodologia, fase e nível de avaliação. A metodologia compreende basicamente as ferramentas típicas empregadas na análise de desempenho de sistemas computacionais : modelagem analítica, simulação e mensuração [KOB 78] (figura 2.4). Cada metodologia é mais adequada para uma determinada fase de observação do sistema (definição, projeto, implementação e operação) e para um nível específico da arquitetura (microeletrônico, registradores, processador-memória, sistema operacional e aplicativo).

A modelagem analítica consiste em desenvolver modelos matemáticos que representam, de uma maneira abstrata e compacta, o comportamento do sistema real. O fundamento teórico comumente empregado para realizar este tipo de modelagem é a teoria de processos estocásticos [KLE 75]. Sob este enfoque, do sistema é abstraído um comportamento regido por regras probabilísticas (variáveis aleatórias) que se modificam com o passar do tempo. A teoria de filas [KLE 75] e as redes de Petri estocásticas [MAR 86] formam um conjunto de resultados derivados da aplicação dos processos estocásticos e são freqüentemente utilizados. A avaliação de desempenho segundo esta abordagem analítica é um processo

rápido, embora os resultados obtidos sejam aproximados, devido às hipóteses simplificadoras assumidas durante este processo. Em geral, modelos matemáticos são aplicados a sistemas ainda não implementados, sobre os quais se deseja obter estimativas "cruas" de seu padrão de comportamento (previsão). Quanto ao nível de uma arquitetura paralela, modelos analíticos são mais adequados para representação de níveis mais altos, tais como processador-memória e sistema operacional, pois estes níveis apresentam um menor número de variáveis efetivamente representativas do seu comportamento.

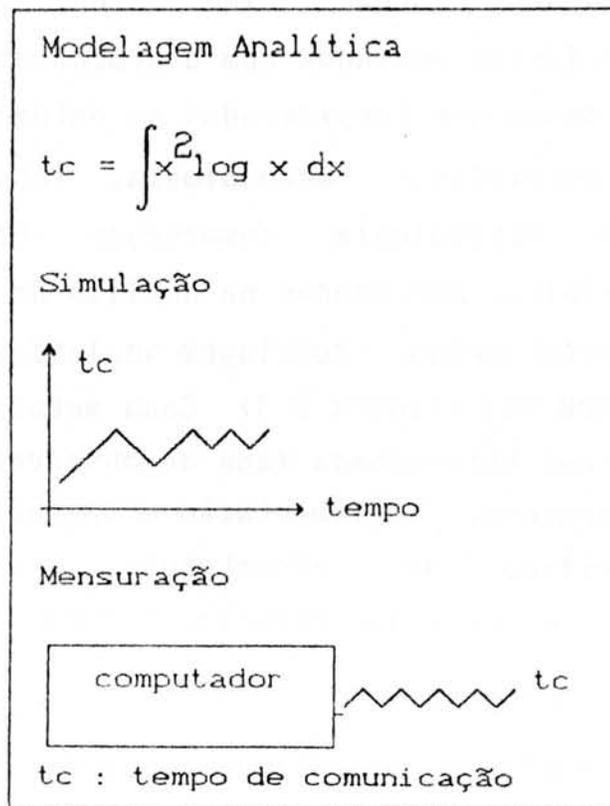


Figura 2.4 - Técnicas de avaliação de desempenho

A **simulação** é também uma forma de modelagem. Porém, o funcionamento do modelo consiste em, não somente representar, mas também repetir o comportamento do sistema real. Em um simulador, a passagem do tempo é também passível de representação e os resultados do modelo são obtidos em função desta passagem do tempo. Em relação à modelagem

analítica, a simulação proporciona índices mais precisos de desempenho. Em contrapartida, o tempo gasto para se realizar a análise é maior. A simulação é empregada em geral para validar os modelos analíticos. Está presente também nas fases iniciais de desenvolvimento de uma máquina paralela. Normalmente, recorre-se a esta abordagem quando o sistema a ser modelado apresenta aspectos de difícil modelagem analítica. Pode-se aplicar a técnica de simulação em qualquer nível de um multiprocessador. Em alguns casos, como para os níveis físicos (microeletrônicos), a simulação é utilizada com um propósito diferente, não associado a desempenho, mas como uma ferramenta de verificação lógica.

A mensuração é tipicamente empregada em sistemas fisicamente implementados. Compreende uma série de técnicas e ferramentas de obtenção de índices diretamente do sistema real : monitoração, "benchmark"(*), análise empírica. Produz resultados fiéis à realidade, mas é dispendiosa em tempo e custo. É também restritiva, pois o acompanhamento é feito em uma configuração fixa. É possível obter medidas de qualquer subsistema de um multiprocessador : há monitores tanto de software como de hardware. Os valores obtidos em uma mensuração podem também ser utilizados para validar modelos analíticos ou de simulação e também servir como entrada para estes modelos.

Cada metodologia de avaliação de desempenho apresenta características e facilidades que permitem a modelagem de subsistemas específicos. Por exemplo, a teoria de filas proporciona resultados coerentes na análise de utilização de recursos compartilhados. Redes de Petri, por outro lado, são interessantes na representação de concorrência e alocação simultânea de recursos. A simulação pode ser utilizada em níveis onde modelos analíticos não são adequados. A mensuração é um meio eficaz de comprovar a validade de todo o método de análise de desempenho. Um estudo ideal de qualquer sistema compõe-se da integração de

todas estas técnicas, integração conhecida como análise de desempenho através de modelos híbridos.

Em resumo, a avaliação de desempenho de máquinas paralelas é importante pois permite quantificar a potência de processamento ganha e garantir que a paralelização de atividades seja uma solução real na busca de maior velocidade de computação. Este processo de análise possibilita a aplicação de uma gama consistente de técnicas e ferramentas e é essencial no desenvolvimento de qualquer sistema, seja este paralelo ou não.

3 MODELAGEM DE SISTEMAS

Em avaliação de desempenho de sistemas computacionais, os modelos possuem um papel preponderante na obtenção de resultados associados ao comportamento destes sistemas. O processo de modelagem empregado neste trabalho inspirou-se em Jacoby [JAC 80], cujas etapas estão ilustradas na figura 3.1.

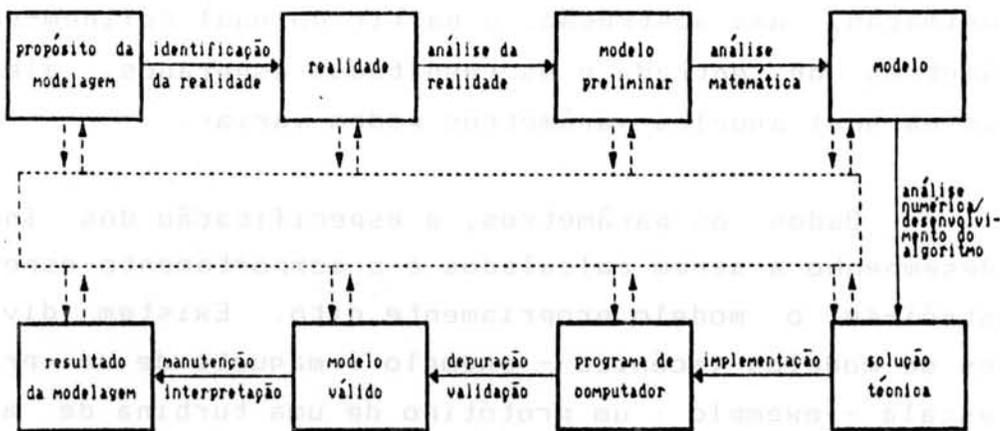


Figura 3.1 - Etapas do processo de modelagem auxiliada por computador [JAC 80]

A fase correspondente ao propósito da modelagem constitui na determinação do porquê da necessidade de um modelo. A modelagem faz parte, em geral, de um projeto maior composto de vários estágios. Por exemplo, a modelagem orientada para escolha de alternativas poderia ser uma etapa de um projeto maior de construção de um computador. A definição do objetivo do modelo dentro deste contexto geral e a especificação dos resultados esperados fazem parte desta fase. O ponto inicial e final da modelagem devem estar bem claros.

Determinada a motivação do modelo, o conceito, o objeto, o sistema ou o processo a ser modelado deve ser devidamente identificado ([JAC 80] chama este elemento de protótipo). A identificação desta realidade e dos aspectos de interesse são o ponto de partida para o desenvolvimento do modelo.

A compreensão da realidade e a síntese dos aspectos de interesse possibilitam a criação de um modelo preliminar. Este modelo despojado já constitui uma aproximação, uma abstração, a partir do qual definem-se os parâmetros de entrada e os resultados esperados, além da faixa na qual aqueles parâmetros podem variar.

Dados os parâmetros, a especificação dos índices de desempenho a serem calculados e o comportamento esperado, constrói-se o modelo propriamente dito. Existem diversos tipos de modelos (icônico - exemplo : maquete de um prédio; de escala - exemplo : um protótipo de uma turbina de avião; gráfico - exemplo : um mapa; matemático - exemplo : a equação correspondente ao teorema de Pitágoras). Para cada tipo, diferentes maneiras de construção de modelos são empregadas. Pode-se utilizar uma teoria, ou mesmo intuição e experiência. Particularmente neste trabalho, o interesse recai sobre modelos matemáticos (analíticos), construídos através de processos fundamentados em teorias matemáticas, e modelos de simulação, os quais permitem uma repetição do comportamento do sistema real.

O modelo matemático é passível de solução a fim de que os índices desejados sejam calculados. Por exemplo, a solução pode ser determinada por simples manipulação algébrica do modelo ou por algum método numérico iterativo, se não for possível encontrar uma forma fechada (algébrica) de resolução do modelo.

O computador é um instrumento essencial de apoio à implementação e à resolução de modelos analíticos, bem como os de simulação numérica. Jacoby [JAC 80] considera o mapeamento do modelo em um computador como uma fase do processo de modelagem. Em geral, este mapeamento resulta em um programa de computador. O responsável pelo desenvolvimento do modelo pode tanto empregar uma linguagem de programação como um pacote aplicativo.

O modelo implementado em computador é capaz de produzir resultados que devem ser validados. A validação consiste em verificar se estes resultados estão de acordo com a realidade modelada, ou seja, se o comportamento do modelo reflete o comportamento desta realidade. Esta comparação deve ser feita sempre que possível diretamente sobre o sistema real, embora nem sempre isto seja possível. A comparação com resultados de outros modelos da mesma realidade é outra forma de validação. O modelo válido pode então ser utilizado a fim de atingir os objetivos definidos na etapa inicial do processo de modelagem. Os resultados obtidos são empregados pelo contexto onde o modelo se insere.

Todas as etapas descritas anteriormente não são isoladas. Existe um mecanismo de retro-alimentação ("feedback"(*)) que permite a correção ou o ajuste do modelo à medida que novos aspectos aparecem durante o processo. Esta natureza iterativa possibilita a continuidade da modelagem e evita o desperdício de tempo e de recursos.

4 COMUNICAÇÃO EM MULTIPROCESSADORES

Dentro da metodologia de modelagem descrita anteriormente, serão apresentadas neste capítulo as três primeiras fases : propósito da modelagem, definição do sistema real e construção de um modelo preliminar (figura 4.1). Alguns aspectos associados à motivação podem ser também encontrados na introdução deste trabalho (capítulo 1).

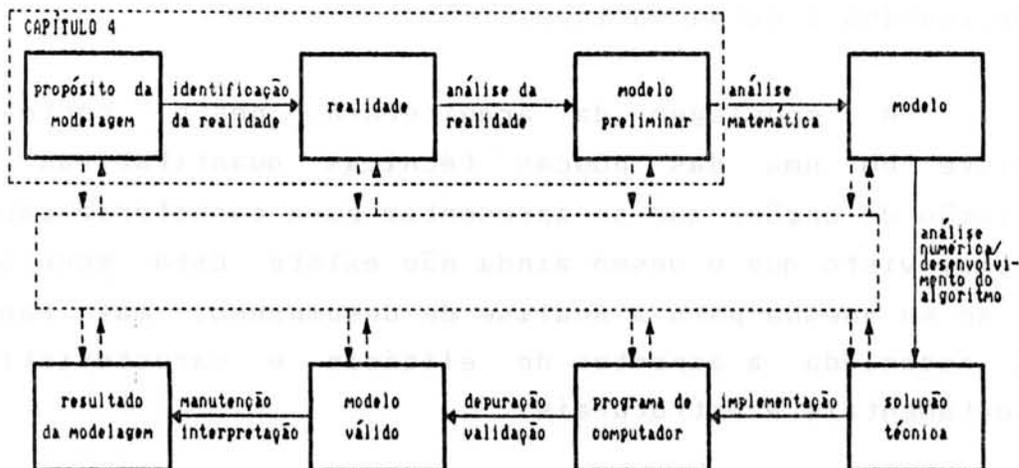


Figura 4.1 - Âmbito do Capítulo 4

4.1 Escopo da modelagem

O desenvolvimento de uma máquina paralela envolve uma série de decisões, dados os objetivos que se espera alcançar com a implementação do sistema proposto. Esta constante escolha por alternativas existe durante todo o período do projeto. Porém, é na fase inicial que estas decisões se tornam "cruciais" e podem ser consideradas "estratégicas". As escolhas realizadas durante a implementação são, por outro lado, "operacionais" e afetam

apenas uma pequena parte de todo o processo (embora, em casos raros, estas pequenas decisões podem conduzir à inviabilidade do projeto).

As considerações que devem ser feitas antes de efetivamente projetar um sistema paralelo se baseiam em informações tais como tipo de aplicações que serão executadas no futuro sistema, tecnologia disponível, tendências do estado da arte, custo, disponibilidade de recursos, tempo de projeto e comportamento de sistemas concorrentes. Todos os níveis da máquina final estão condicionados a estas variáveis.

A modelagem da arquitetura, neste contexto, consiste em uma das poucas técnicas quantitativas de avaliação de opções que se apresentam para caracterização do sistema, visto que o mesmo ainda não existe. Esta modelagem não só se adequa para a análise de desempenho, mas também está associada a aspectos de eficácia e características comportamentais e estruturais.

4.2 Comunicação

A máquina paralela de interesse para este trabalho é o multiprocessador fracamente acoplado, composto de nós processadores interligados por uma rede de interconexão arbitrária. Não há memória comum e cada processador possui autonomia de computação (figura 2.2).

Cada ligação entre dois processadores é composta de dois canais : um de saída e outro de entrada, que operam de maneira independente. A multiplexação, ou seja, a divisão do canal físico em mais de um canal virtual, não é considerada. As ligações podem funcionar simultaneamente. A conectividade de um processador é determinada pelo número de ligações de saída acopladas àquele processador.

A troca de dados entre os nós é feita através de mensagens, compostas de duas partes : um cabeçalho e os dados propriamente ditos. O cabeçalho possui informações de controle como o endereço do nó destinatário (figura 4.2). Todos os bits de uma mensagem são transmitidos sem interrupção. A comunicação é assíncrona. A mensagem, para chegar ao seu destino, pode atravessar mais de um processador, considerados como intermediários. Os módulos paralelos, localizados em processadores distintos, dependem destas mensagens para trocar dados que necessitam. As mensagens tornam-se elementos de sincronização e um sistema eficiente de comunicação entre processadores deve existir para garantir a transmissão rápida e eficaz destas mensagens.

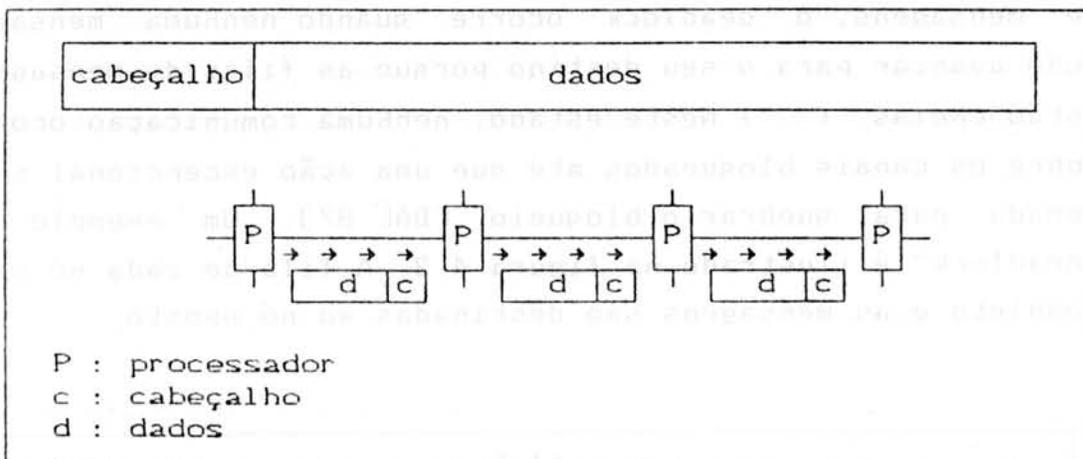


Figura 4.2 - Formato de uma mensagem

A comunicação eficiente em um multiprocessador depende de aspectos físicos, como capacidade de transferência de bits de cada canal, tamanho de áreas de armazenamento de mensagens e características topológicas da rede, e de aspectos lógicos, tais como procedimentos de controle da rede, como o roteamento, cujas decisões permitem

especificar qual a seqüência de canais que a mensagem deve percorrer até chegar ao seu destino final. A obtenção desta seqüência não é um processo trivial. O roteamento pode ser estático, quando esta seqüência permanece fixa durante toda a execução dos módulos de um programa; ou dinâmico, se as decisões de encaminhamento de mensagens se alteram, segundo o estado corrente do multiprocessador. O roteamento dinâmico é mais difícil de se pôr em prática, pois constantemente todos os nós devem estar a par do nível de congestionamento de todos os canais do multiprocessador. A sobrecarga da comunicação pode também ser minimizada se o algoritmo paralelo for adequadamente projetado.

Os algoritmos de roteamento devem prevenir também a ocorrência de "deadlocks"(*) (livremente traduzido como "bloqueio mortal"). Em um multiprocessador baseado em troca de mensagens, o "deadlock" ocorre "quando nenhuma mensagem pode avançar para o seu destino porque as filas de mensagens estão cheias. (...) Neste estado, nenhuma comunicação ocorre sobre os canais bloqueados até que uma ação excepcional seja tomada para quebrar o bloqueio" [DAL 87]. Um exemplo de "deadlock" é ilustrado na figura 4.3. A fila de cada nó está completa e as mensagens são destinadas ao nó oposto.

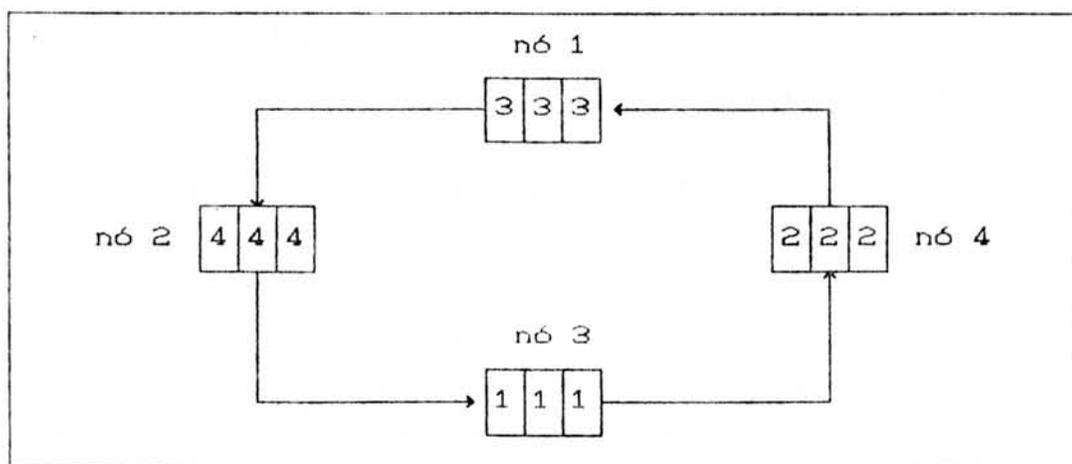


Figura 4.3 - Bloqueio mortal [DAL 87]

Um parâmetro de análise do grau de comunicação em um multiprocessador é o tempo médio de comunicação entre dois processos, localizados em nós distintos. Este intervalo de tempo abrange desde o momento em que o processador emissor envia a mensagem até a recepção completa da mesma pelo destinatário (figura 4.4). Este tempo pode compôr uma fração considerável do tempo total necessário para executar um algoritmo.

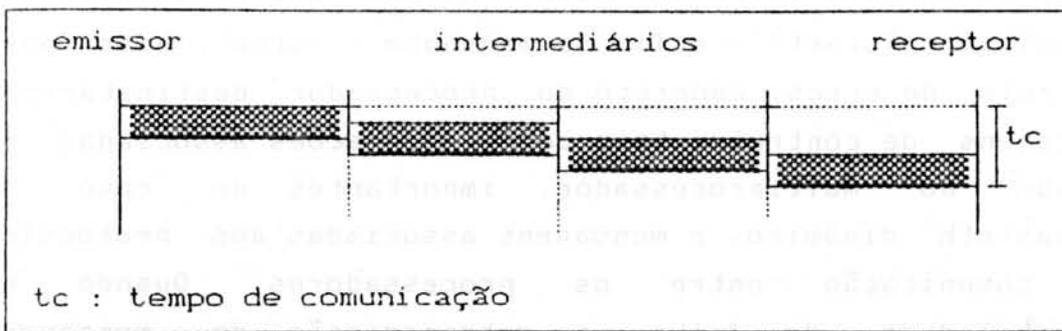


Figura 4.4 - Tempo de comunicação

A comunicação envolve uma série de atrasos que se somam [BER 89] :

- tempo de processamento : necessário para preparar a informação a ser transmitida, obter informações de roteamento e realizar procedimentos de controle de erros
- tempo de enfileiramento : tempo gasto na espera por liberação de recursos necessários à comunicação, como as linhas de comunicação
- tempo de transmissão : correspondente ao tempo gasto para a transmissão (descarga) de todos os bits da mensagem pelo canal de comunicação, não incluindo o tempo de propagação da linha

- tempo de propagação : intervalo de tempo entre o fim da transmissão do último bit da mensagem no processador origem e a recepção do último bit desta mensagem pelo processador vizinho.

Em particular, quanto ao tempo de comunicação, a taxa efetiva de transmissão de bits através dos canais é quase sempre menor do que a taxa nominal, pois há uma quantidade de bits que circulam pela rede que não estão relacionados com os processos do algoritmo da solução. Por exemplo, os bits extras de cada mensagem (bit de sinalização de início - "start" - e fim de mensagem - "stop", bits para controle de erros, endereço do processador destinatário), mensagens de controle, tais como informações associadas ao estado do multiprocessador, importantes no caso de roteamento dinâmico, e mensagens associadas aos protocolos de comunicação entre os processadores. Quando há possibilidade de falha, a retransmissão de mensagens contribui para também diminuir a capacidade de transmissão dos canais.

O modelo preliminar da comunicação entre dois processadores é visualizado na figura 4.5. Neste modelo, supõe-se que a comunicação é feita de forma totalmente confiável e que os algoritmos de roteamento sempre encontram um caminho entre o processador origem e o destino (não há tempo de processamento nos processadores). As estratégias de comunicação garantem a não ocorrência de bloqueios mortais ("deadlock"). Os únicos atrasos considerados são o de transmissão e o de enfileiramento (espera de liberação de linha de transmissão). O parâmetro avaliado é o tempo de comunicação total entre os dois processadores. A conectividade de um processador é sempre maior do que a unidade (não existem nós terminais na rede). A rede é considerada no equilíbrio (em estado de regime) e aquelas mensagens que não são consumidas pelo processador destino (ou seja, estão de passagem por um nó intermediário) são

igualmente distribuídas entre as ligações de saída daquele nó. Todas as premissas acima adotadas são necessárias durante a modelagem a fim de que se possa chegar a modelos de fácil resolução.

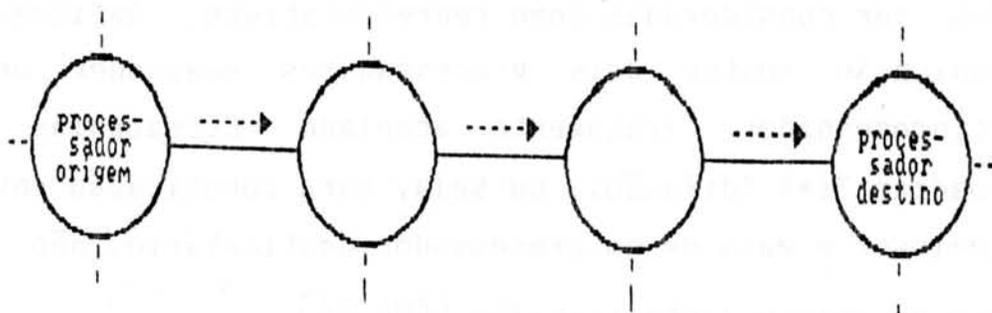


Figura 4.5 - Modelo preliminar de comunicação

4.3 Estratégias de comunicação

A comunicação entre dois processos situados em processadores fisicamente distintos pode ocorrer de diversas maneiras, de acordo com estratégias diferentes. Quatro destas estratégias foram escolhidas para análise: a comutação de mensagens ("message switching"(*)) [KER 79], o "virtual cut-through"(*) [KER 79], o "rendez-vous"(*) [PLA 89] e o "wormhole"(*) [DAL 87]. A escolha acima não foi arbitrária. A comutação de mensagens é um método empregado em redes de computadores e que pode também ser implementado em multiprocessadores. O "virtual cut-through" se assemelha à comutação de mensagens, mas explora a possibilidade de utilização de dispositivos físicos dedicados ao roteamento. Processos descritos em certas linguagens (exemplo: Occam [TRY 89]) trocam dados segundo um protocolo denominado "rendez-vous". Este protocolo é válido para troca de mensagens entre processadores vizinhos. O "rendez-vous" pode ser adaptado para a comunicação entre dois processadores quaisquer da rede, se se deseja conservar a mesma estratégia (a nível de programa) independente da localização dos nós. O "wormhole" também depende de dispositivos dedicados ao

roteamento, assim como o "virtual cut-through". Porém, a particularidade do "wormhole" é o aproveitamento de um certo paralelismo existente durante a troca de mensagens.

Estas quatro estratégias não são as únicas, mas podem ser consideradas como representativas. Aplicam-se à comunicação entre dois processadores quaisquer de um multiprocessador fracamente acoplado. Estratégias para "broadcast"(*) (difusão), ou seja, para comunicação entre um nó emissor e mais de um processador destinatário, não fazem parte do escopo deste trabalho [CHA 84].

4.3.1 A estratégia comutação de mensagens

As descrições desta e das outras estratégias baseiam-se no modelo preliminar da figura 4.5. O processador emissor envia uma mensagem a um nó determinado da rede. Esta mensagem poderá percorrer um ou mais canais de comunicação até chegar ao seu destino. Um processador intermediário, ao receber uma mensagem, analisa o cabeçalho da mesma e verifica o endereço de destino. Normalmente, uma acusação de recebimento (ACK - "Acknowledgement"*) é enviada ao processador vizinho emissor da mensagem. A acusação de recebimento de mensagem é importante principalmente se há alguma exigência quanto à confiabilidade da comunicação, mas não é considerada neste estudo.

Analisado o cabeçalho, o processador verifica se ele próprio é o destino. Caso positivo, a comunicação está encerrada. Caso este processador seja apenas um intermediário, ele decide, baseado em informações de roteamento, por qual canal de saída a mensagem será transmitida. Tomada esta decisão, a mensagem é colocada em uma fila. Quando a linha estiver livre, a mensagem é transmitida, recebida pelo processador vizinho e novamente analisada. Esta rotina é repetida até a mensagem chegar ao

processador destino (figura 4.6).

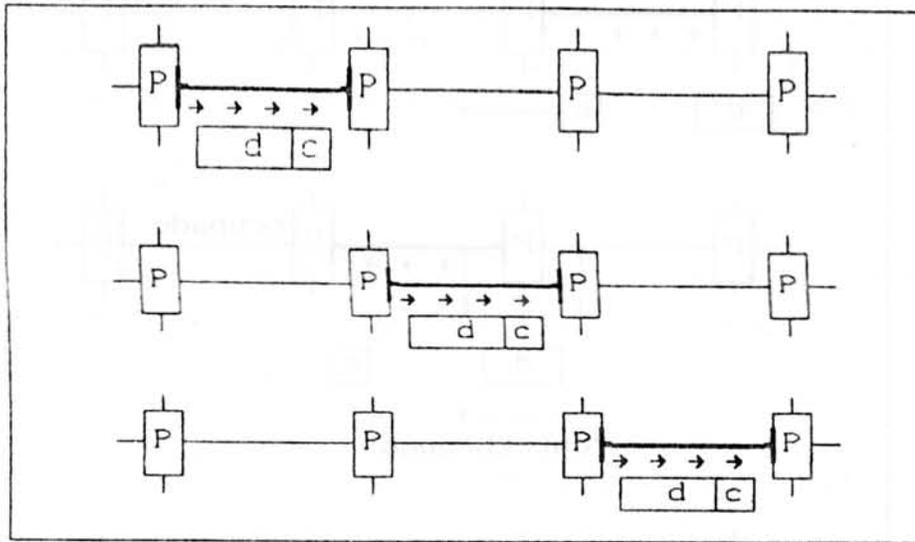
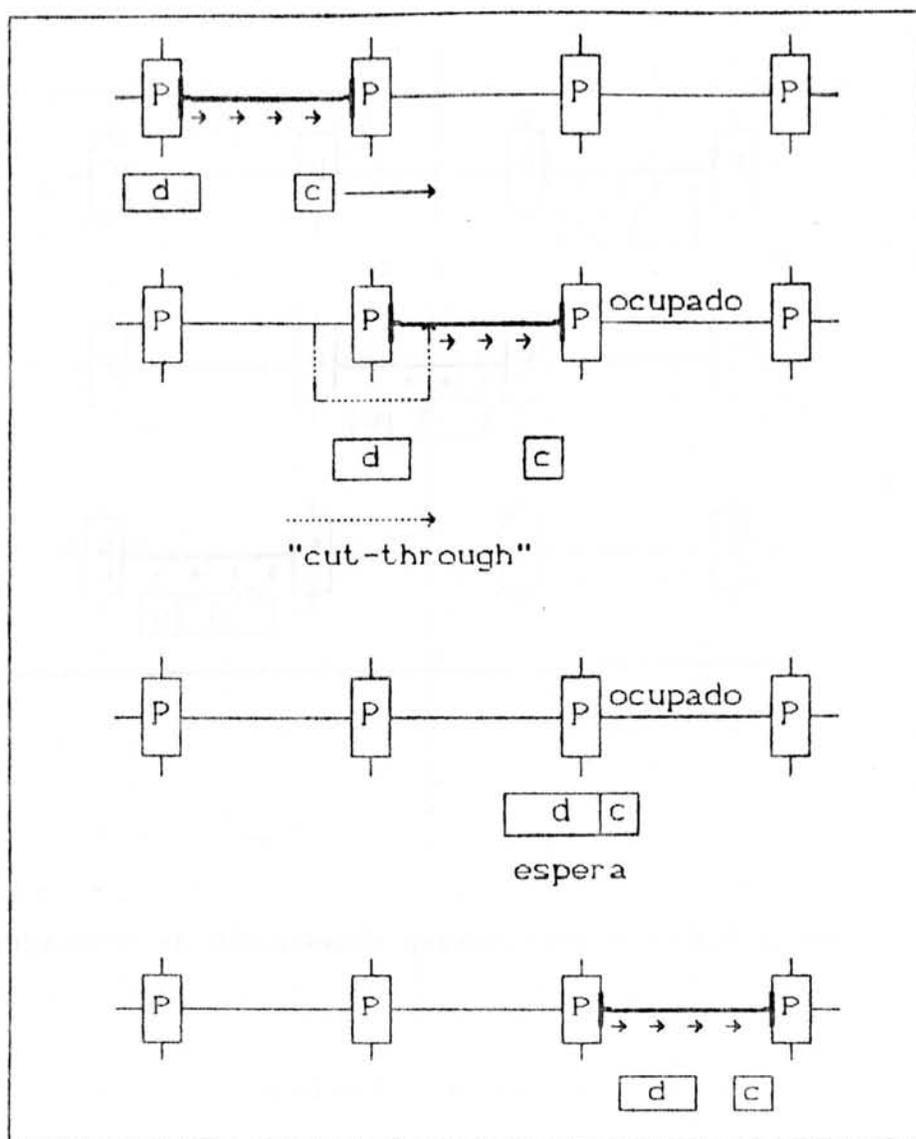


Figura 4.6 - A estratégia comutação de mensagens

4.3.2 A estratégia "virtual cut-through"

O "virtual cut-through" ("cut through" é uma expressão da língua inglesa que significa "atravessar algo ligeiramente") é muito semelhante à comutação de mensagens. A única diferença está no momento em que a mensagem é recebida por um processador intermediário. O cabeçalho da mesma é analisado em tempo de recepção, antes mesmo da mensagem restante ser armazenada. Para tanto, é necessário existir um dispositivo físico dedicado a esta análise. Decodificado o endereço de destino, o roteamento determina um canal de saída e se o mesmo estiver livre, a mensagem é automaticamente direcionada para esta linha de saída, não ocorrendo retransmissão da parte de dados. O armazenamento no nó intermediário com a colocação da mensagem na fila só

ocorre se o canal estiver ocupado (figura 4.7).



(Hipótese : canais 1 e 2 livres/canal 3 ocupado)

Figura 4.7 - A estratégia "virtual cut-through"

4.3.3 A estratégia "rendez-vous"

O "rendez-vous" (expressão da língua francesa que significa "encontro, reunião") utiliza um procedimento semelhante à comutação de circuitos, adotado pela telefonia. O processador que deseja comunicar-se envia uma solicitação de comunicação ao processador destino. Esta solicitação é

conduzida pela rede da mesma forma que uma mensagem é enviada pelo processo de comutação. A diferença é que, a cada linha percorrida, o canal é reservado, ou seja, nenhuma outra comunicação pode utilizá-lo. Se, durante o percurso, algum canal estiver alocado para outra comunicação, a solicitação aguarda pela liberação do mesmo. No momento em que a solicitação de comunicação chega ao nó destino, o mesmo envia (pelos canais já reservados) um aceite de comunicação. O emissor, ao receber o aceite, envia a mensagem propriamente dita. A mensagem percorre o caminho reservado e libera o mesmo, à medida que atravessa a rede (figura 4.8).

4.3.4 A estratégia "wormhole"

O "wormhole" (expressão da língua inglesa que denota "buraco de minhoca ou de bicho") também apresenta características de reserva de canais. A mensagem a ser enviada é particionada no emissor em segmentos indivisíveis denominados "flits" (*) ("flow control digits" - dígitos de controle de fluxo). Os processadores transmitem "flits" que podem ser de cabeçalho ou de dados. Uma mensagem pode conter um ou mais "flits" de cabeçalho e um ou mais "flits" de dados. O processador emissor (sempre de acordo com regras ditadas pelo roteamento) determina um canal de saída. Com o canal livre, os "flits" da mensagem são enviados. Ao chegar em um nó intermediário, os "flits" de cabeçalho são analisados por dispositivos dedicados ao roteamento. Se a linha escolhida estiver livre, os "flits" são automaticamente comutados. Caso contrário, somente os "flits" de cabeçalho são armazenados no nó à espera da liberação do canal. Os demais ficam distribuídos nos nós já percorridos. Os canais entre o nó que contém o cabeçalho e o que contém o último "flit" estão todos reservados, pois os segmentos de dados não contêm informações de roteamento. É possível que o primeiro "flit" da mensagem chegue no destino

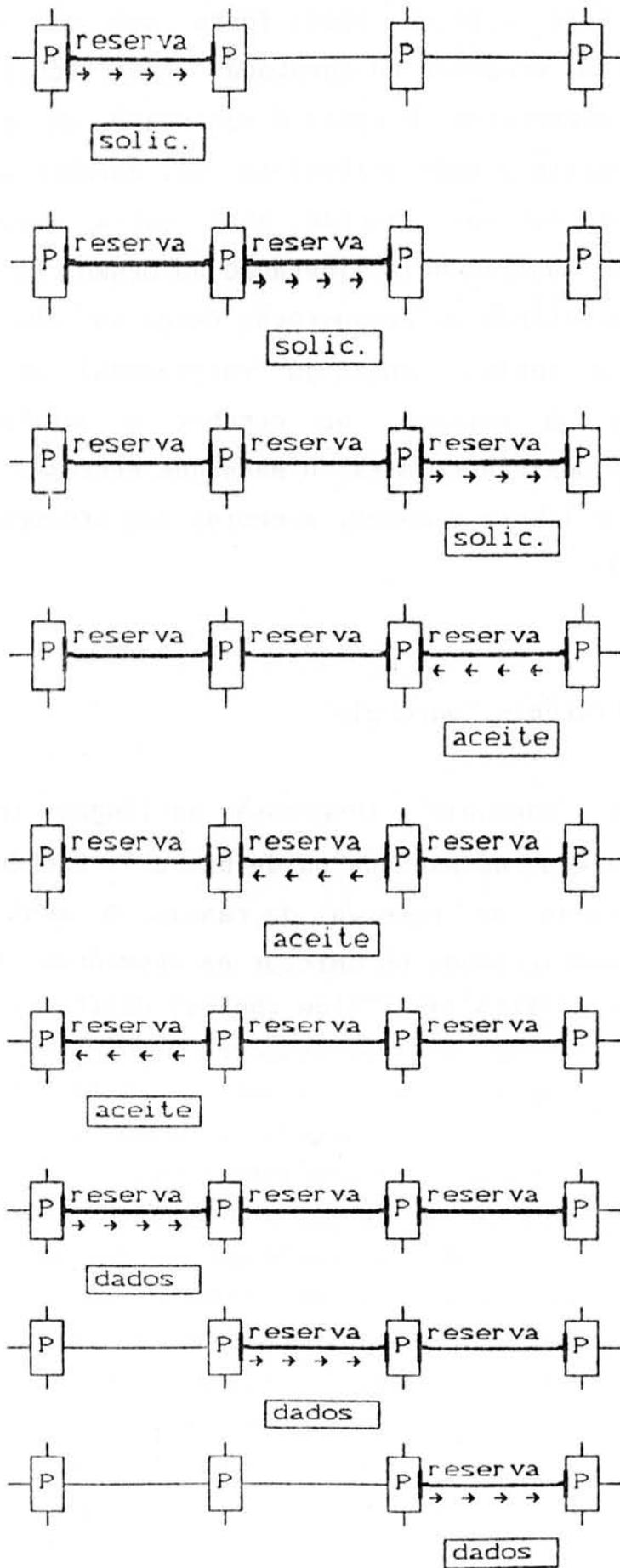
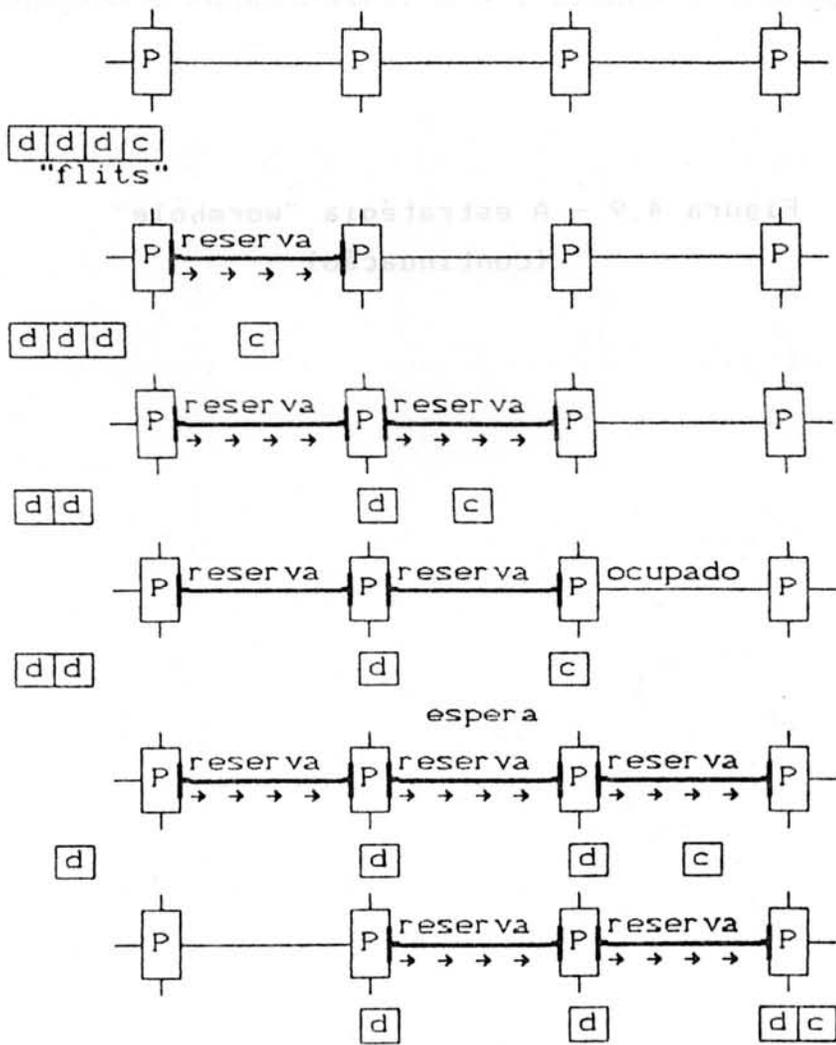


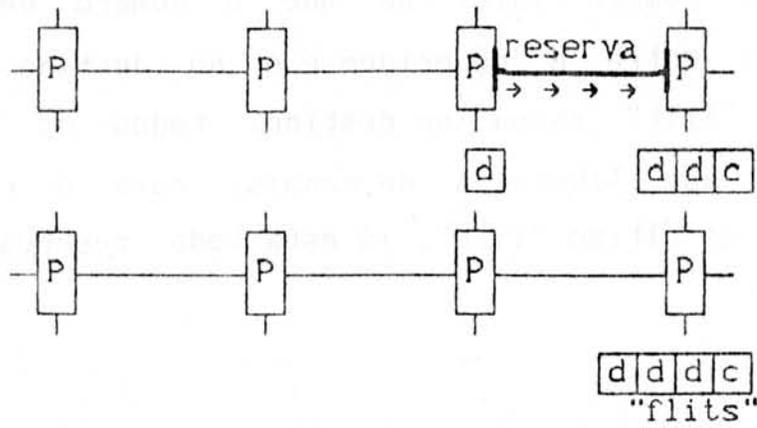
Figura 4.8 - A estratégia "rendez-vous"

antes do último ser enviado pelo processador emissor. Neste trabalho, considera-se sempre este caso : o número de "flits" é sempre maior do que o número de ligações existentes entre o nó origem e o nó destino. Quando o primeiro "flit" chega no destino, todos os demais não esperarão por liberação de canais, pois o caminho, do primeiro ao último "flit", já está todo reservado (figura 4.9).



(Hipótese : canais 1 e 2 livres/canal 3 ocupado)

Figura 4.9 - A estratégia "wormhole"
(continua)



(Hipótese : canais 1 e 2 livres/canal 3 ocupado)

Figura 4.9 - A estratégia "wormhole"
(continuação)

5 MODELOS ANALÍTICOS

Este capítulo e o próximo concernem às fases seguintes do processo da modelagem: o desenvolvimento do modelo e a determinação do método de resolução (figura 5.1).

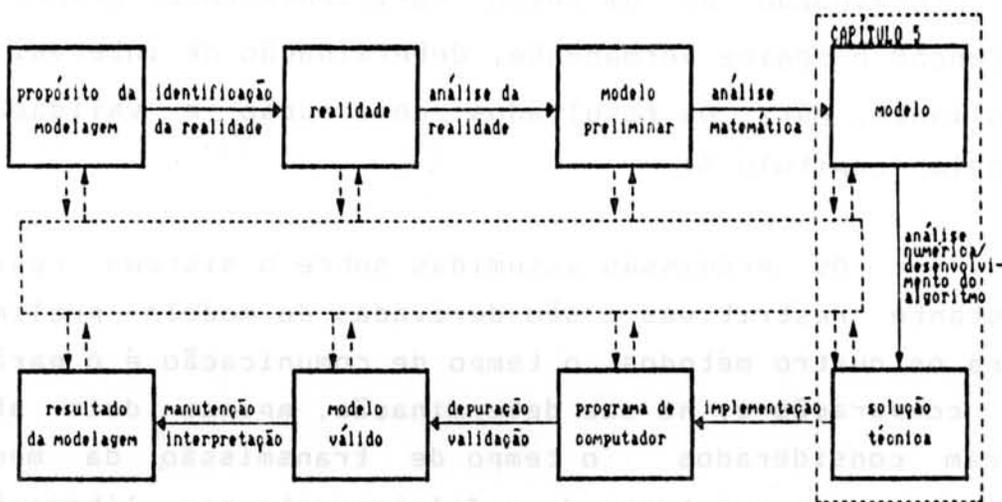


Figura 5.1 - Âmbito do capítulo 5

As estratégias descritas no capítulo 4 podem ser adotadas em diferentes configurações de multiprocessadores e sob diferentes cargas de trabalho. A análise de desempenho de cada método pode ser feita através de modelos analíticos, simulação ou através da mensuração direta de um sistema real. Como analisado anteriormente, a última opção apresenta uma série de inconvenientes, tais como a disponibilidade da máquina e o custo de analisar o comportamento da arquitetura sob diferentes cargas de trabalho. Visto que, aqui, o propósito da modelagem é auxiliar no projeto de arquiteturas, a mensuração não se enquadra neste contexto. Os modelos analíticos, por outro lado, são compactos, ainda que imprecisos pois consideram somente os aspectos mais importantes do sistema. Todavia, estudos feitos com modelos

analíticos têm apresentado resultados satisfatórios. Modelos foram validados e, sob determinadas condições, os resultados calculados têm estado de acordo com a realidade [SCH 67]. A simulação é outra ferramenta para análise de desempenho. Mais custosa do que a modelagem analítica e mais flexível do que a mensuração, a simulação consiste em repetir o comportamento do sistema real. Os problemas da simulação são a determinação de um tempo suficientemente grande para alcançar o regime permanente, determinação de intervalos de confiança para os resultados, construção e validação do modelo (capítulo 6).

As premissas assumidas sobre o sistema real são bastante restritivas e são derivadas do modelo preliminar. Para os quatro métodos, o tempo de comunicação é o parâmetro de comparação e, na sua determinação, apenas dois atrasos foram considerados : o tempo de transmissão da mensagem pelos canais e o tempo de enfileiramento por liberação de linhas de transmissão. O roteamento é considerado adequado, não há ocorrência de "deadlocks". Os componentes do multiprocessador são totalmente confiáveis. A topologia da rede é arbitrária. Com estas premissas, os modelos procuram enfocar o mecanismo para a efetivação da comunicação. Nos sistemas reais, tempos de processamento (roteamento, controle de erros) e tempos de propagação existem, mas podem ser desprezados se forem considerados estes atrasos como constantes.

A técnica utilizada nos quatro casos para modelagem analítica é a teoria de filas básica. No anexo 1, podem ser encontrados maiores detalhes sobre os fundamentos desta teoria. A modelagem analítica através desta ferramenta (e não, por exemplo, através de redes de Petri) justifica-se pela tentativa de aproximar um sistema de filas bloqueantes (processadores que se comunicam por "rendez-vous" ou "wormhole") por um sistema clássico de filas, onde a reserva de recursos não existe. A validação desta aproximação,

frente à simulação, é realizada no capítulo 7. O "rendez-vous" e o "wormhole" foram modelados analiticamente neste trabalho. As estratégias não bloqueantes (comutação de mensagens e "virtual cut-through") já tinham sido adequadamente modeladas com a teoria de filas [KER 79].

5.1 Modelo analítico : comutação de mensagens

Este modelo, desenvolvido por Kermani e Kleinrock em [KER 79], considera as seguintes hipóteses (baseadas no modelo preliminar) :

- a distribuição do tempo entre chegadas de mensagens ao sistema é exponencial
- o comprimento da mensagem segue uma distribuição exponencial
- a capacidade do nó para armazenar mensagens é infinita
- o roteamento é determinístico. O canal de saída não é escolhido ao acaso
- os componentes da rede (processadores e canais) são independentes uns dos outros
- a rede está balanceada, ou seja, o fator de utilização de todos os canais é o mesmo
- as ligações são confiáveis (não há necessidade de envio de confirmação de recebimento de mensagem).

Os parâmetros do modelo (figura 5.2) são a taxa média de chegada de mensagens (λ), o comprimento médio das mensagens (L), a capacidade média do canal (a sua taxa de transmissão, representado por C) e o número médio de "hops" (*), ou seja, o número médio de canais a serem percorridos (n_h).

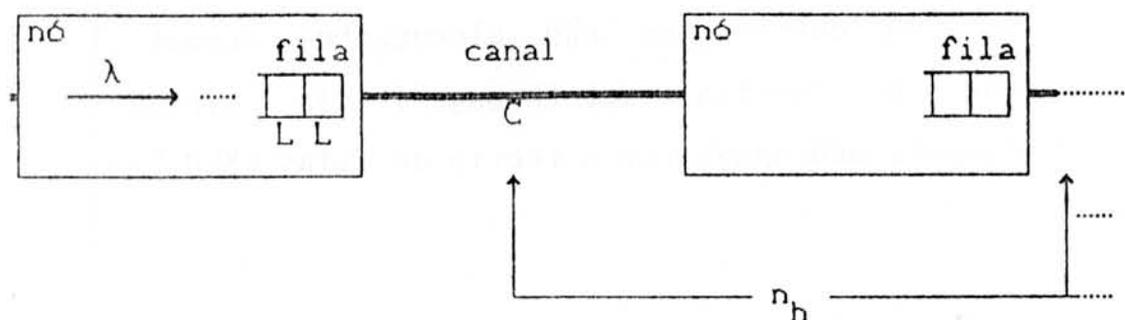


Figura 5.2 - Modelagem da comutação de mensagens

Cada canal pode ser considerado como um servidor, ou seja, ele realiza uma tarefa que consiste em transmitir mensagens de um processador a outro. As mensagens são os elementos que impõem uma demanda sobre o servidor. Quando o canal está ocupado, as mensagens esperam. De acordo com este comportamento e com as premissas descritas acima, o modelo de um canal pode ser representado por um sistema de filas M/M/1 (as distribuições do tempo entre chegadas de clientes e do tempo de serviço, na fila M/M/1, são exponenciais e há somente um servidor - vide anexo 1). O tempo de passagem por este sistema é composto de dois períodos: o tempo de espera mais o tempo de serviço. O tempo médio de serviço (TMS) corresponde ao tempo de transmissão da mensagem pelo canal.

$$\text{TMS} = \frac{L}{C} \quad (1)$$

O tempo médio de espera (TME) é derivado a partir da análise do sistema M/M/1, considerando que TMS segue uma distribuição exponencial.

$$TME = \frac{\lambda TMS^2}{1 - \lambda TMS} \quad (2)$$

Como o número médio de canais a serem percorridos é n_h , o tempo médio total de comunicação é dado por :

$$TC = (TMS + TME) n_h = \left[\frac{L}{C} + \frac{\lambda (L / C)^2}{1 - \lambda (L / C)} \right] n_h \quad (3)$$

5.2 Modelo analítico : "virtual cut-through"

Esta estratégia também foi modelada por Kermani e Kleinrock em [KER 79]. O mesmo modelo preliminar e as mesmas premissas da comutação de mensagens valem também para o "virtual cut-through". O parâmetro extra a ser considerado é a razão entre o tempo de transmissão do cabeçalho e o tempo total de transmissão da mensagem (alfa).

Para cada "cut-through" realizado, um período de transmissão de mensagem é economizado, dado que o tempo de espera foi nulo. Mas como o "cut-through" corresponde a uma transmissão do cabeçalho e não dos dados, efetivamente o ganho corresponde ao número de "cut-throughs" multiplicado pelo tempo de transmissão da parte de dados. Mas o número de "cut-throughs" é dado pela equação (4), onde $(\lambda * (L / C))$ é a utilização do canal. O fator $(1 - (\lambda * (L / C)))$ corresponde à probabilidade de ocorrer um "cut-through" (probabilidade do sistema M/M/1 estar vazio).

$$n_c = (n_h - 1) (1 - \lambda(L / C)) \quad (4)$$

Logo a diferença entre o tempo de comunicação na comutação de mensagens e o tempo de comunicação com "cut-through" é dado por :

$$T_m - T_c = (n_h - 1)(1 - \lambda(L / C))(1 - \alpha)t_m \quad (5)$$

Que se reduz a :

$$T_c = T_m - (n_h - 1)(1 - \lambda(L / C))(1 - \alpha)t_m \quad (6)$$

5.3 Modelo analítico : "rendez-vous"

A reserva de um caminho entre o processador emissor e o receptor e a espera de outras solicitações pela liberação deste percurso tornam a modelagem desta estratégia um tanto complexa.

O tempo médio de comunicação entre dois processadores constitui uma soma de atrasos de transmissão e de espera : tempo de transmissão da solicitação (t_{s01}), tempo médio de espera da solicitação em cada processador

pela liberação de canal (W), tempo de transmissão do aceite de comunicação (t_{ack}) e tempo de transmissão da mensagem (t_{msg}). Primeiro, a solicitação percorre a rede e gasta um tempo de espera em cada nó ($n_h * (t_{sol} + W)$). O aceite é então enviado ($n_h * t_{ack}$). Finalmente, a mensagem percorre a rede ($n_h * t_{msg}$).

$$T = (t_{sol} * n_h) + (W * n_h) + (t_{ack} * n_h) + (t_{msg} * n_h) \quad (7)$$

Os atrasos de transmissão podem ser calculados a partir do tamanho das mensagens trocadas entre os processadores e a capacidade das linhas de transmissão. Todo o problema está em determinar W , ou seja, o tempo de espera enfrentado pela solicitação a fim de obter um canal de comunicação.

O tempo de espera deve ser calculado a partir das seguintes variáveis :

- tempo médio de serviço do canal (TMS) que, neste caso, é denominado de tempo de alocação ("holding time"(*))
- taxa de chegada de mensagens (λ) que necessitam do canal para fazer a comunicação.

O "holding time" é definido como a média aritmética do tempo de alocação das várias comunicações que utilizam aquele canal (figura 5.3). Cada nó é um gerador de mensagens internas. O canal que precede aquele nó na comunicação também é um gerador, mas de mensagens externas. A demanda por uma linha é assim composta de duas sub-demandas : uma gerada pelo próprio nó e outra proveniente da ligação anterior.

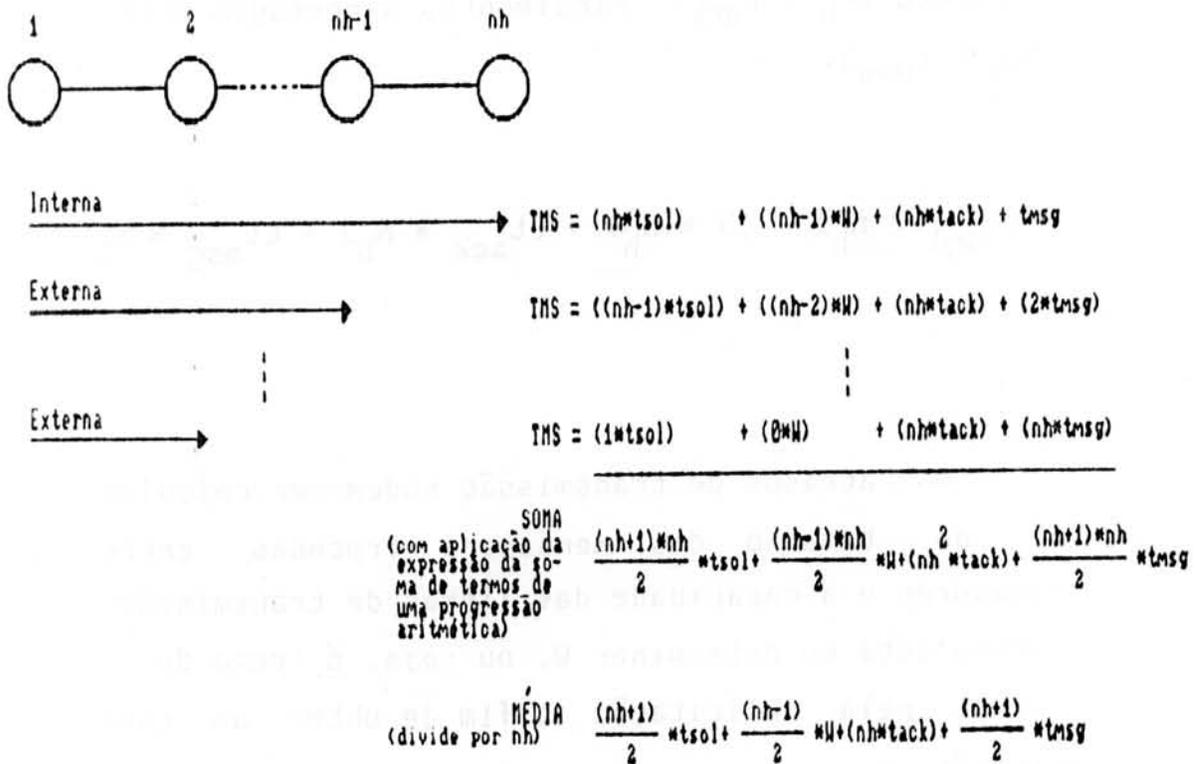


Figura 5.3 - Tempo médio de alocação de canal no "rendez-vous"

O tempo de espera é calculado segundo a expressão do tempo de espera de um sistema M/M/1 (anexo 1). O tempo entre chegadas de mensagens é o parâmetro de variação e o tempo de serviço corresponde ao "holding time" (tempo de alocação). A distribuição do tempo entre chegada de mensagens é aproximada por uma exponencial. A determinação da distribuição real que rege estes tempos é complexa, devido à dependência entre as mesmas: o tempo de espera depende do tempo de alocação, que, por sua vez, possui W na sua expressão. A capacidade de armazenamento de um nó é infinita.

Conseqüentemente, a relação entre o "holding time" e o tempo de espera é representado por um sistema de duas equações a duas variáveis :

$$\left\{ \begin{array}{l} \text{TMS} = \frac{(n_h+1)}{2} t_{\text{sol}} + \frac{(n_h-1)}{2} W + n_h t_{\text{ack}} + \frac{(n_h+1)}{2} t_{\text{msg}} \\ W = \frac{\lambda * \text{TMS}^2}{1 - \lambda * \text{TMS}} \end{array} \right. \quad (8)$$

O método de resolução do sistema acima pode ser algébrico ou iterativo. Calculado o valor de W, o mesmo é substituído na equação (7).

5.4 Modelo analítico : "wormhole"

O "wormhole", assim como o "rendez-vous", é também uma estratégia difícil de ser modelada, pois apresenta situações de bloqueio e de reserva. Embora os modelos sejam diferentes, o desenvolvimento é análogo.

O tempo de comunicação (T) entre dois processadores é dado pela equação abaixo. O número médio de "flits" da mensagem é dado por n_f e o tempo de transmissão do "flit" corresponde a T_f . No início, um "flit" (de cabeçalho) percorre a rede com esperas nos processadores intermediários ($n_h * (T_f + W)$). O caminho já está todo reservado : os demais "flits" podem ser transmitidos de uma só vez ($(n_f - 1) * T_f$).

$$T = n_h (T_f + W) + (n_f - 1) T_f \quad (9)$$

Como no "rendez-vous", o problema também é determinar o tempo médio de espera (W) enfrentado pelos "flits" em cada processador intermediário. Este tempo médio de espera é determinado a partir de um tempo de serviço virtual do canal, que corresponde a um tempo de alocação de cada canal ou "holding time". Como no "rendez-vous", este tempo é considerado como a média aritmética do tempo de alocação das comunicações que reservam aquela ligação (comunicações externas ou oriundas do processador emissor) (figura 5.4).

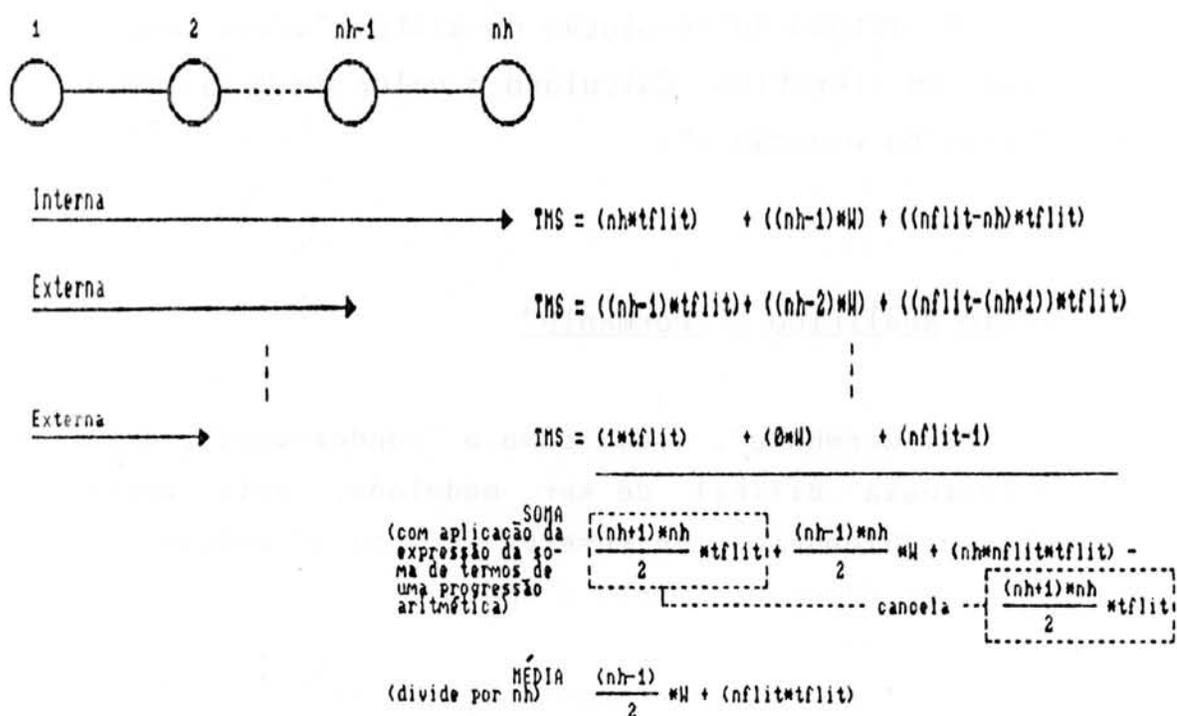


Figura 5.4 - Tempo médio de alocação de canal no "wormhole"

O modelo consiste em um sistema de equações (nas variáveis TH e W) e a determinação de W pode ser feita por um método algébrico ou iterativo, análogo ao modelo do "rendez-vous". A hipótese exponencial para a geração de mensagens internamente em um nó é mantida.

As expresões do "holding time" e do tempo de espera são dadas pelo sistema abaixo :

$$\left\{ \begin{array}{l} \text{TMS} = \frac{(n_h - 1)}{2} W + (n_{\text{flit}} * t_{\text{flit}}) \\ \\ W = \frac{\lambda \text{TMS}^2}{1 - \lambda \text{TMS}} \end{array} \right. \quad (10)$$

O W calculado é substituído na equação (9).

THE UNIVERSITY OF CHICAGO

PHYSICS DEPARTMENT

PHYSICS 311

PROBLEM SET 10

6 MODELOS DE SIMULAÇÃO

A simulação também é uma ferramenta para modelagem e análise de sistemas. Neste capítulo, serão apresentados os modelos de simulação para o "rendez-vous" e para o "wormhole". Para a comutação de mensagens e para o "virtual cut-through", não serão desenvolvidos modelos para simulação, pois estas estratégias já foram devidamente modeladas e validadas na literatura [KER 79]. O modelo analítico apresentou-se coerente. Alguns gráficos da validação de [KER 79] são apresentados no capítulo 7. Para compreender integralmente este capítulo, é necessário conhecer GPSS. O anexo 2 apresenta conceitos básicos desta ferramenta de simulação. Os anexos 3 e 4 apresentam os programas fontes das simulações realizadas.

Analogamente ao capítulo 5, as fases do processo de modelagem aqui abrangidas neste capítulo estão ilustradas na figura 6.1.

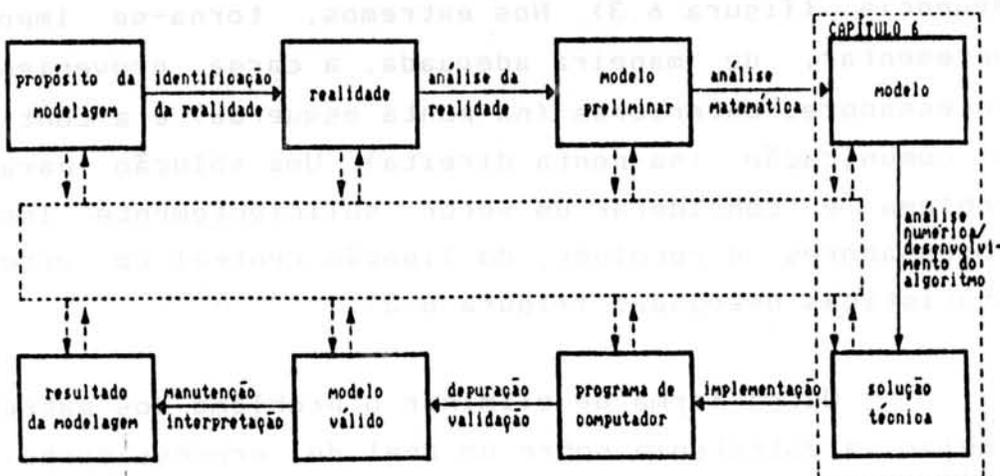


Figura 6.1 - Âmbito do capítulo 6

6.1 Modelo de simulação : "rendez-vous"

A construção do modelo da simulação se baseou na disponibilidade do sistema simulador GPSS(*) ("General Purpose System Simulator" - vide anexo 2). A partir do modelo preliminar apresentado no capítulo 4 e, segundo algumas premissas adotadas na modelagem analítica no capítulo 5, o modelo para a simulação (em um anel) é apresentado na figura 6.2.

A linguagem GPSS e os blocos do fluxo são explicados no anexo 2.

O tempo entre geração de mensagens em um nó segue uma distribuição exponencial. O número de "hops" ou linhas a serem percorridas é considerado constante, já que, no modelo analítico, este parâmetro é considerado fixo e nenhuma informação sobre sua distribuição é revelada. A topologia modelada se constitui em uma seqüência linear e finita de processadores. O problema desta configuração para a modelagem do bloqueio é a influência das pontas desta seqüência (figura 6.3). Nos extremos, torna-se impossível representar, de maneira adequada, a carga proveniente de processadores anteriores (na ponta esquerda) e a continuação da comunicação (na ponta direita). Uma solução para este problema é considerar um vetor suficientemente longo de processadores e recolher, da ligação central do vetor, as estatísticas desejadas (figura 6.3).

Outra forma de eliminar o problema dos extremos é simular a estratégia sobre um anel de processadores, onde não existem pontas. A desvantagem do anel é o efeito que uma comunicação pode eventualmente ter sobre ela mesma, principalmente nas situações próximas da saturação, visto que (a) um processador afeta o próximo nó da comunicação e (b) o anel constitui um ciclo fechado (figura 6.4). Serão analisadas as duas topologias, a título de comparação.

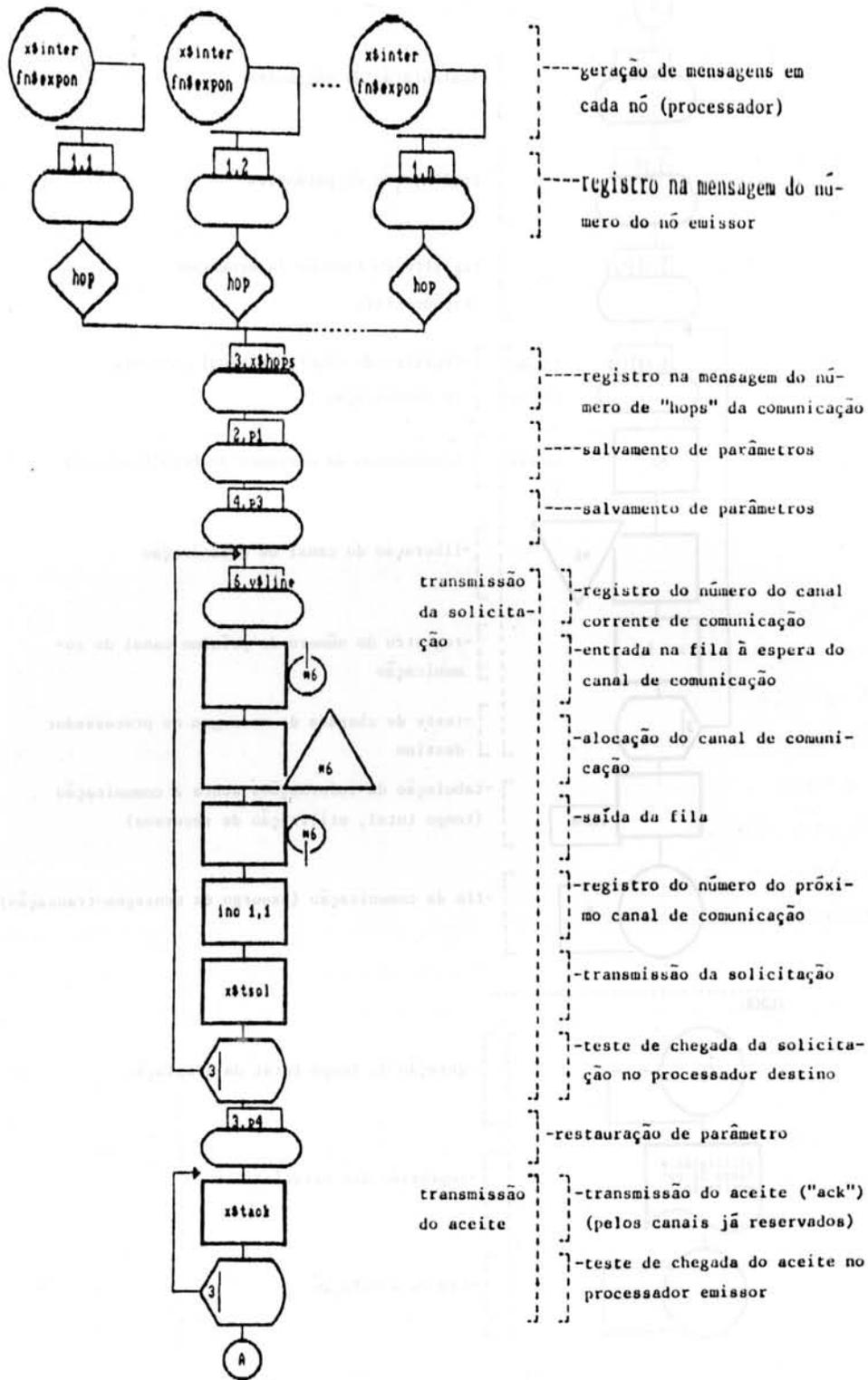


Figura 6.2 - Fluxograma do modelo de simulação do "rendez-vous" (continua)

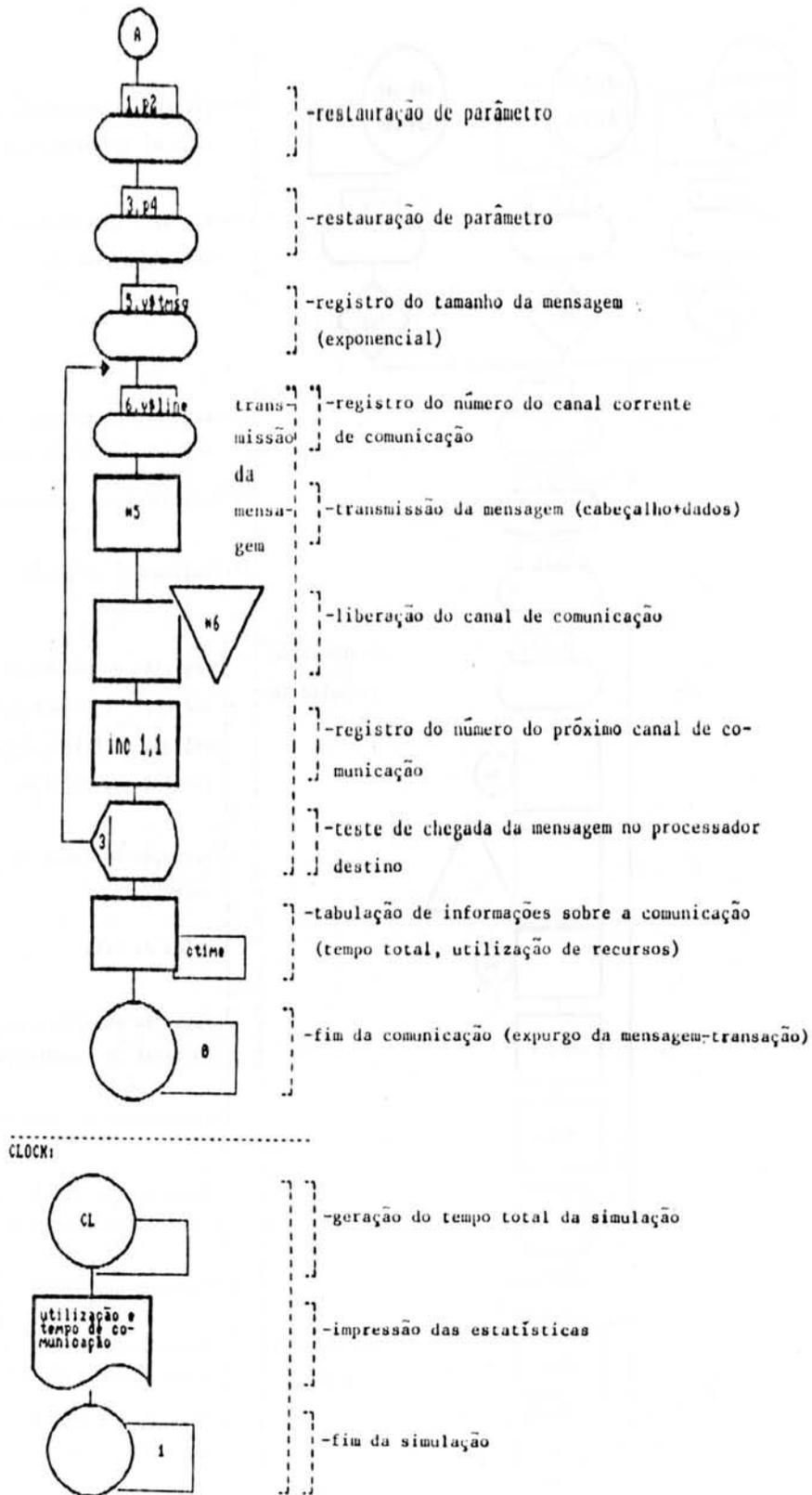


Figura 6.2 - Fluxograma do modelo de simulação do "rendez-vous" (continuação)

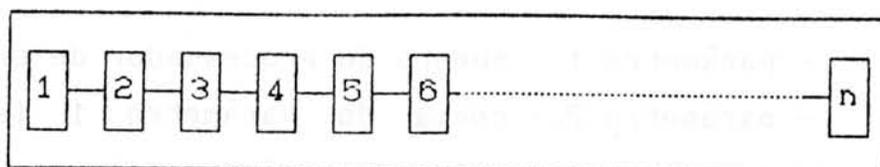


Figura 6.3 - Sequência linear de processadores

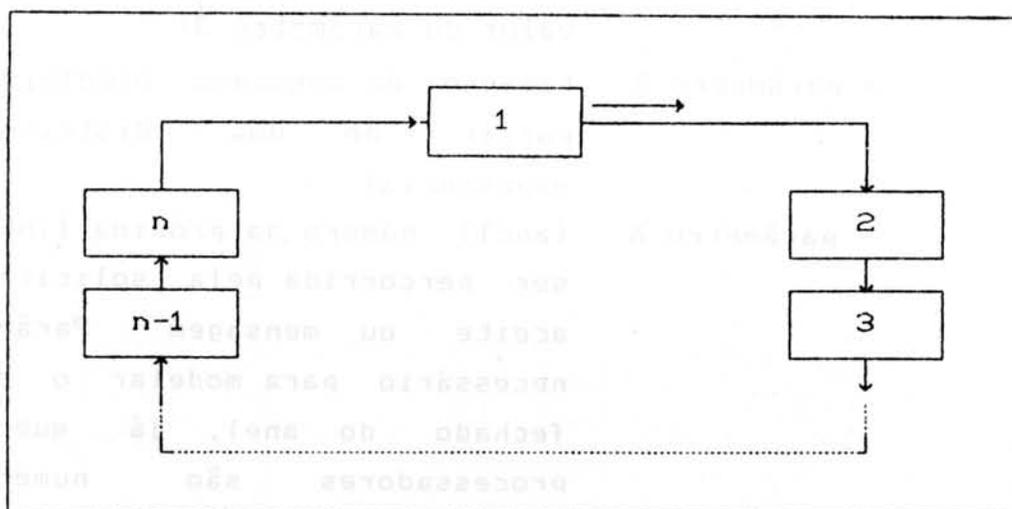


Figura 6.4 - Anel de processadores

A escolha de uma topologia (seqüência linear, anel ou qualquer outra) na simulação é uma questão delicada, pois os modelos analíticos não consideram as características topológicas do multiprocessador e o simulador não possibilita a implementação de um modelo com topologia "arbitrária". A escolha de uma seqüência linear se baseou em [KER 79], onde o autor enfrenta o mesmo problema de escolha de topologia. A configuração em anel foi implementada a fim de contornar a situação particular que ocorre nos extremos da seqüência (situação descrita no parágrafo anterior).

O modelo simulado é composto de cinco partes. Cada transação (uma transação corresponde a uma mensagem) possui cinco parâmetros (no caso do vetor de processadores) ou seis

(no caso do anel) :

- parâmetro 1 : número do processador de origem
- parâmetro 2 : cópia do parâmetro 1 (salva o valor do parâmetro 1)
- parâmetro 3 : número de "hops" a percorrer para aquela transação
- parâmetro 4 : cópia do parâmetro 3 (salva o valor do parâmetro 3)
- parâmetro 5 : tamanho da mensagem escolhido a partir de uma distribuição exponencial
- parâmetro 6 : (anel) número da próxima linha a ser percorrida pela solicitação, aceite ou mensagem. Parâmetro necessário para modelar o ciclo fechado do anel, já que os processadores são numerados (figura 6.4). É o resultado da aplicação da função módulo.

Os parâmetros fornecidos pelo usuário são (a capacidade de transmissão da linha é sempre um bit por unidade de tempo) :

- X\$INTER : média do tempo entre a geração de mensagens
- X\$HOPS : número de "hops" da comunicação. No caso do vetor, este valor é menor para os processadores da ponta direita (figura 6.3)
- X\$TSOL : tempo para transmitir uma solicitação por uma linha
- X\$TACK : tempo para transmitir um aceite (ACK) por uma linha
- X\$TMSG : média do tempo para transmitir uma mensagem (distribuição exponencial)

CLOCK : tempo da simulação.

O fluxo pode ser subdividido em cinco partes :

1. geração das mensagens nos processadores
2. transmissão da solicitação (primeiro laço)
3. transmissão do aceite pelos canais já reservados (segundo laço)
4. transmissão da mensagem (terceiro laço)
5. geração do tempo da simulação.

A diferença entre a simulação em um anel e a simulação em um vetor está somente na existência do parâmetro 6. No anel, a numeração das facilidades corresponde a um ciclo : utiliza-se a função módulo.

O método de análise estatística dos resultados da simulação foi o da execução única ("single run") [KOB 78]. Nesta técnica, o modelo é simulado até que um estado de equilíbrio seja atingido. A determinação deste ponto é feita através de simulações piloto, nas quais o equilíbrio é determinado através da variação da utilização das linhas de transmissão. Considera-se estável a utilização da linha quando a variação da mesma não ultrapassa 10 %. Por exemplo, simula-se por um período X supostamente longo suficiente para englobar o equilíbrio. Divide-se esta simulação em intervalos fixos de tempo. Ao final de cada intervalo, coleta-se somente a estatística desejada. Se este valor se estabilizar (daí a variação menor do que 10 %), o equilíbrio é considerado como alcançado e a sua duração pode ser conferida a partir do relatório produzido pelo simulador.

Determinado o comprimento do estado transiente do sistema, simula-se o mesmo por este período, zeram-se as

estatísticas (as transações correntes são mantidas), simula-se por mais um período equivalente ao do equilíbrio, zeram-se novamente as estatísticas e assim por diante. Este procedimento se repete mais vezes, tantas quantas forem as amostras necessárias. A cada vez, o sistema simulador coleta o valor da utilização de cada ligação e o tempo médio de comunicação (tempo de trânsito das transações). A partir do conjunto de amostras, determina-se o intervalo de confiança dos dois índices, através da equação abaixo, onde m é a média do levantamento e s corresponde ao desvio padrão. Esta expressão corresponde à técnica de determinação de intervalos de confiança através da utilização da distribuição de Student para n amostras [KOB 78].

$$m \pm t_{(1-\alpha);n} \sqrt{\frac{s^2}{n}}$$

m = média das amostras
 α = grau de confiabilidade
 $(0 < \alpha < 1)$
 n = número de amostras
 s^2 = desvio padrão (11)
 $t_{(1-\alpha);n}$ = valor da distribuição de Student com parâmetros $(1-\alpha)$ e n

O método da execução única apresenta o inconveniente das amostras serem dependentes entre si, já que o estado final de um lote de simulação corresponde ao estado inicial do lote seguinte. [KOB 78] afirma, contudo, que se a duração de cada lote for suficientemente grande, esta interdependência torna-se desprezível.

No anel, os parâmetros obtidos a partir da simulação são o tempo de comunicação entre processadores e o índice de utilização dos canais. No vetor, devido ao problema das pontas, o parâmetro recuperado é o tempo de

espera de transações no processador central do vetor e o grau de utilização do canal central, onde a influência dos extremos é minimizada.

6.2 Modelo de simulação : "wormhole"

A simulação da estratégia "wormhole" se assemelha à simulação do "rendez-vous". O mesmo método de análise estatística de dados foi aplicado. O fluxo da simulação é ilustrado na figura 6.5. Simula-se em um anel e em um vetor linear de processadores.

Cada transação no sistema apresenta os seguintes parâmetros :

- parâmetro 1 : número do processador de origem
- parâmetro 2 : cópia do parâmetro 1 (salva o valor do parâmetro 1)
- parâmetro 3 : número de "hops" a percorrer para aquela transação
- parâmetro 4 : cópia do parâmetro 3 (salva o valor do parâmetro 3)
- parâmetro 5 : tamanho da mensagem (em "flits") obtido a partir de uma distribuição exponencial. Na fase final da simulação, contém o número de "flits" da mensagem que faltam ser transmitidos até a liberação da primeira linha
- parâmetro 6 : (anel) número da próxima linha a ser percorrida pelo primeiro "flit". Parâmetro necessário para modelar a transmissão pelo ciclo do anel. Aplicação da função módulo.

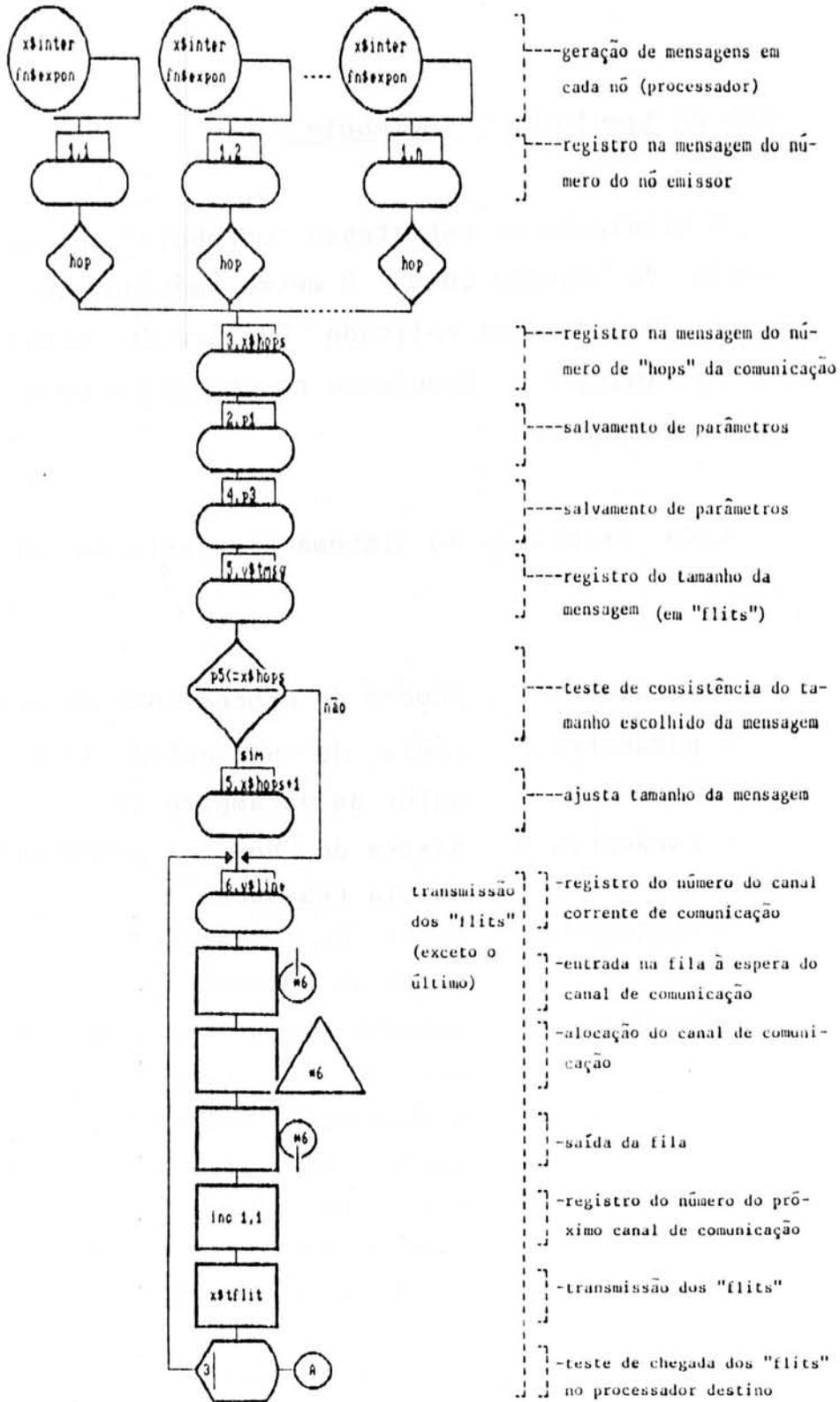


Figura 6.5 - Fluxograma do modelo de simulação do "wormhole" (continua)

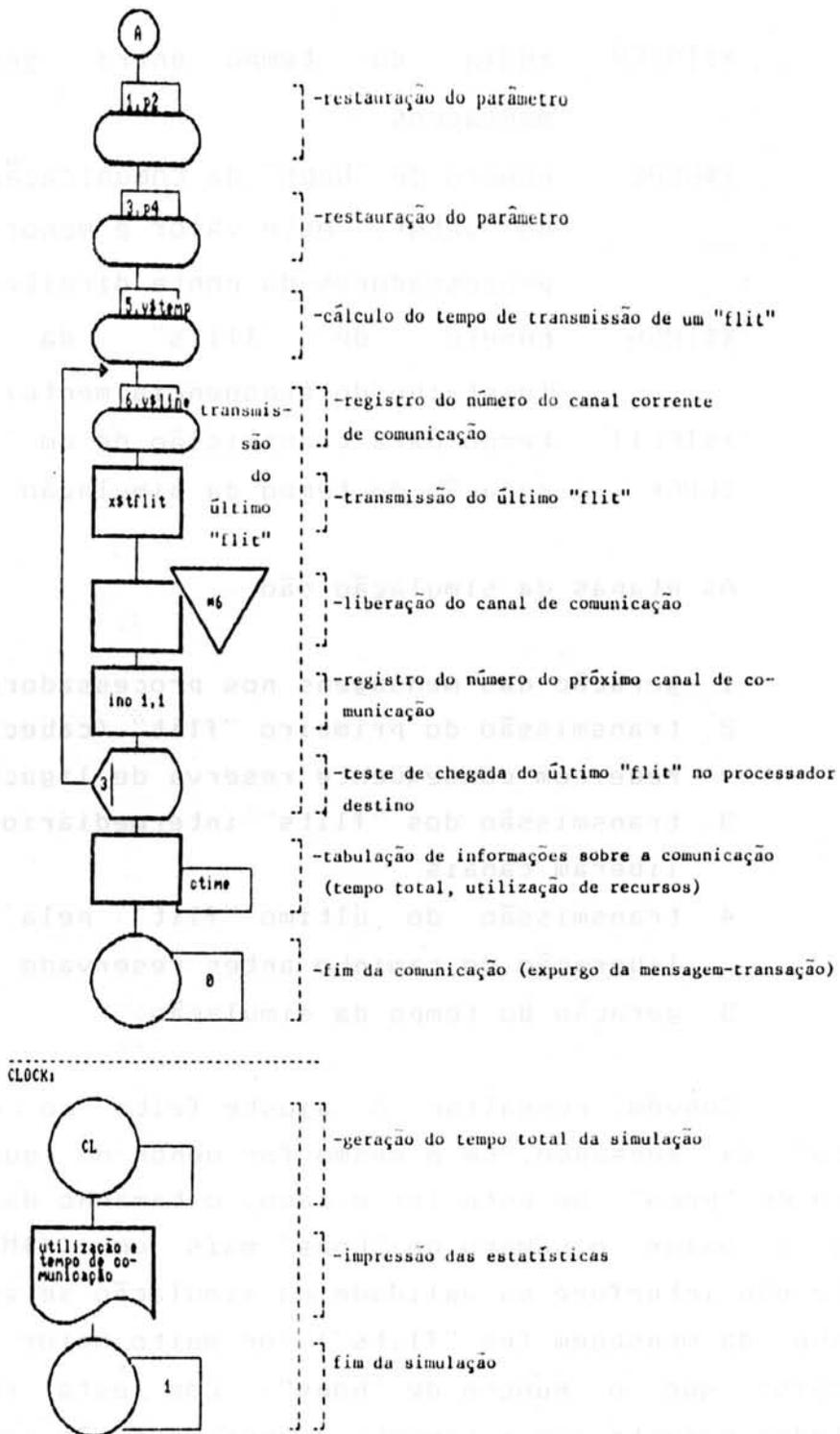


Figura 6.5 - Fluxograma do modelo de simulação do "wormhole" (continuação)

Os parâmetros fornecidos pelo usuário são :

X\$INTER : média do tempo entre geração de mensagens
X\$HOPS : número de "hops" da comunicação. No caso do vetor, este valor é menor para os processadores da ponta direita
X\$TMSG : número de "flits" da mensagem (distribuído exponencialmente)
X\$TFLIT : tempo para transmissão de um "flit"
CLOCK : geração do tempo da simulação.

As etapas da simulação são :

1. geração das mensagens nos processadores
2. transmissão do primeiro "flit" (cabeçalho) pela rede com conseqüente reserva de ligações
3. transmissão dos "flits" intermediários que não liberam canais
4. transmissão do último "flit" pela rede com liberação do caminho antes reservado
5. geração do tempo da simulação.

Convém ressaltar o ajuste feito no número de "flits" da mensagem, se o mesmo for menor ou igual que o número de "hops". Se este for o caso, o tamanho da mensagem passa a valer o número de "hops" mais um (X\$MIN). Este ajuste não interfere na validade da simulação se a média do tamanho da mensagem (em "flits") for muito maior (50 vezes ou mais) que o número de "hops". Com esta relação, o simulador garante que o tamanho da mensagem não será pequeno (a probabilidade do "sorteio exponencial" de um tamanho pequeno de mensagem é reduzida) [DON 79].

ANÁLISE DOS MODELOS

Neste capítulo, serão abordadas as três últimas fases do processo de modelagem: a implementação em um computador, a validação e a utilização do modelo (figura 7.1).

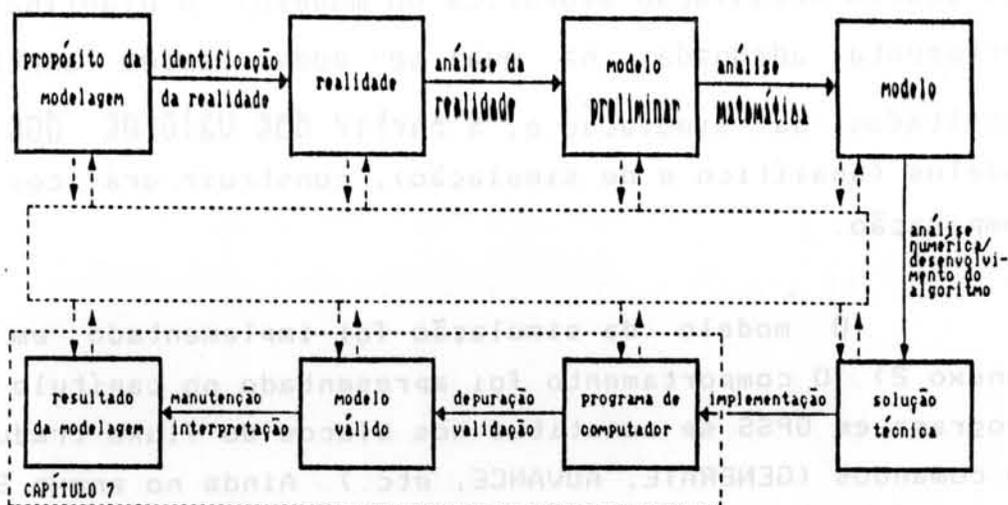


Figura 7.1 - Âmbito do capítulo 7

7.1 Implementação dos modelos

O parâmetro de avaliação do grau de comunicação em um multiprocessador fracamente acoplado é o tempo médio total para efetivação desta comunicação (capítulo 5). Em todos os modelos apresentados, este tempo é calculado em função do índice de utilização média em um canal típico, índice este que varia de 0 a 1. A razão da escolha deste parâmetro se deve a sua facilidade de obtenção, tanto no modelo analítico como, principalmente, no simulado. Nas expressões matemáticas, a utilização é dada pelo produto entre λ (taxa de chegada de mensagens para utilização do canal) e o tempo médio de alocação daquele canal. Na

simulação, este valor é fornecido automaticamente nos relatórios, sob o item "utilização das facilidades".

O modelo analítico foi implementado na planilha de cálculos Lotus 1-2-3 [WIL 86]. Em uma coluna da planilha, foram digitados os valores possíveis da utilização dos canais (de 0 a 1, com passos de 0,01). Em outra coluna, calculou-se o tempo de comunicação em função desta utilização (resolução algébrica do modelo). A planilha é uma ferramenta adequada, na qual se pode também anotar os resultados da simulação e, a partir dos valores dos dois modelos (analítico e de simulação), construir gráficos para comparação.

O modelo da simulação foi implementado em GPSS (anexo 2). O comportamento foi apresentado no capítulo 6 e o programa em GPSS se constitui nos blocos do fluxo traduzidos em comandos (GENERATE, ADVANCE, etc.). Ainda no anexo 2, há exemplos dos programas de simulação das estratégias "rendez-vous" e "wormhole".

7.2 Validação

Na figura 7.2, são apresentados os gráficos de comparação de topologias (anel x vetor). Foram simuladas dez topologias : um anel com 10, 15, 20, 25 e 30 processadores e um vetor com 10, 15, 20, 25 e 30 processadores. Ambas as estratégias foram testadas com os seguintes parâmetros :

- "Rendez-vous" : - 4 "hops"
- tamanho da solicitação : 4 bits
- tamanho do ACK : 1 bit
- tamanho da mensagem : 5000 bits
- capacidade da linha : 1 bit/unidade de tempo
- tempo entre geração de men-

sagens : 750.000 unidades de tempo

- intervalo de coleta de estatísticas : 100.000 unidades de tempo

"Wormhole" : - 4 "hops"

- tamanho do "flit" : 5 bits
- número de "flits" : 1001
- capacidade da linha : 1 bit/unidade de tempo
- tempo entre geração de mensagens : 750.000 unidades de tempo
- intervalo de coleta de estatísticas : 100.000 unidades de tempo.

Os modelos foram simulados até a obtenção do estado de equilíbrio. No anel, foram coletadas as utilizações do canal número 5. A escolha deste canal foi aleatória, já que no estado de regime, todos os canais apresentam a mesma utilização. No vetor, coletaram-se as utilizações do canal central (no vetor de 10 processadores, por exemplo, o canal central corresponde ao quinto canal; no vetor de 15, ao sétimo canal e assim por diante).

Pelos gráficos, observa-se que ambas as topologias são equivalentes. O anel, porém, apresenta um inconveniente : o estado de saturação ocorre antes do que no vetor. Isto é normal, dado que o ciclo do anel propicia à formação de "deadlocks". Se, por exemplo, há uma mensagem em cada nó, o sistema está mutuamente bloqueado. Em situações de saturação (utilização próxima de 1), o número de processadores do vetor apresenta influência.

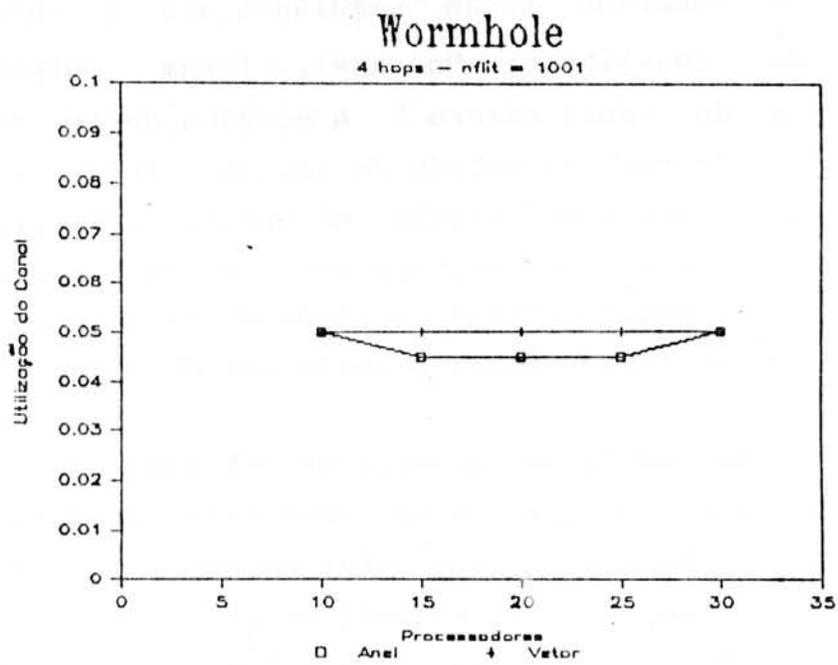
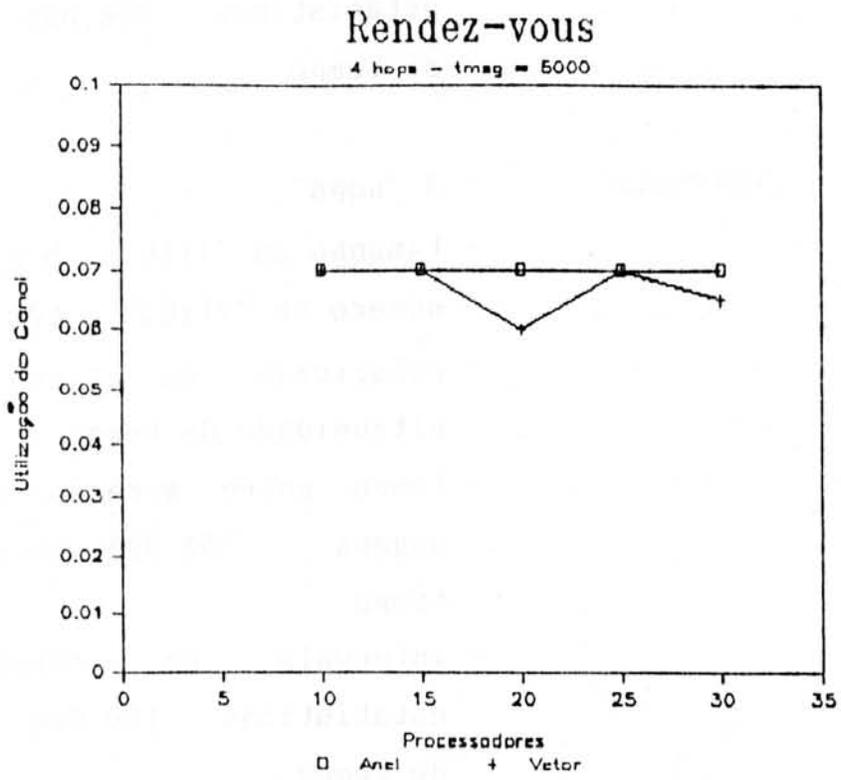


Figura 7.2 - Resultados da comparação entre topologias

Na figura 7.3, são apresentados os gráficos da validação do "cut-through" realizada por [KER 79]. O modelo por ele descrito considera casos de multiplexação, indicado pelo parâmetro N_{ch} (número de canais virtuais). A modelagem da comutação de mensagens é considerada correta, baseada na própria validade da teoria de filas. [KER 79] não apresenta resultados de simulação para esta estratégia.

Na figura 7.4, encontram-se os gráficos correspondentes à validação do modelo analítico contra resultados da simulação. As coordenadas exatas dos pontos dos gráficos estão no anexo 3. Os parâmetros considerados foram :

- "Rendez-vous" :
- 2 e 3 "hops"
 - tamanho da solicitação : 4 bits
 - tamanho do ACK : 1 bit
 - tamanho da mensagem : 1000, 5000 e 10000 bits
 - capacidade da linha : 1 bit/unidade de tempo
 - vetor de 10 processadores (2 "hops") e de 15 processadores (3 "hops").
- "Wormhole"
- 2 e 3 "hops"
 - tamanho do "flit" : 5 bits
 - número de "flits" : 201, 1001 e 2001
 - capacidade da linha : 1 bit/unidade de tempo
 - vetor de 10 processadores (2 "hops" e 3 "hops").

Na simulação, consideraram-se várias taxas de geração de mensagens, com o objetivo de se obter diferentes utilizações de canal. Os parâmetros de saída do modelo foram

$\bar{n}_h = 2.71$

resultados da simulação analítico

\diamond $N_{ch} = 1$
 \circ $N_{ch} = 2$
 \triangle $N_{ch} = 4$

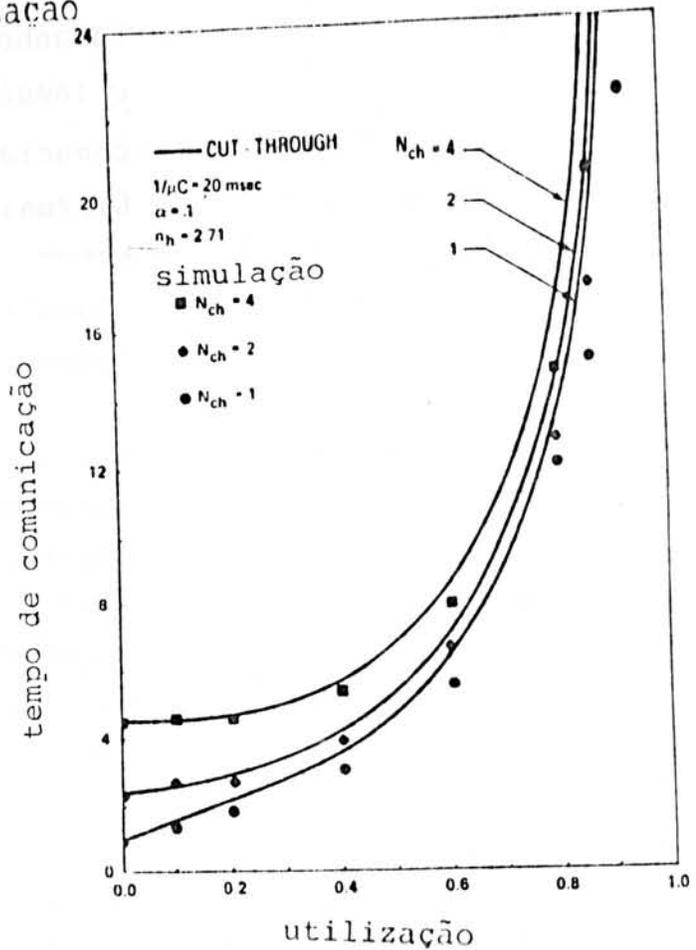
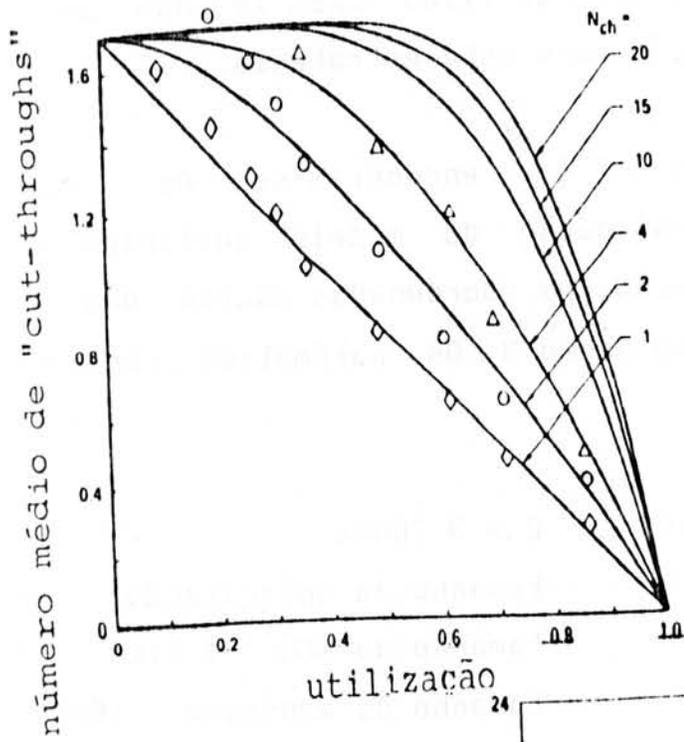


Figura 7.3 - Validação do "cut-through"

a utilização do canal central e o intervalo de tempo que as mensagens geradas pelo processador central levam para chegar ao seu destino (tempo de comunicação).

7.3 Análise das estratégias

Os modelos analíticos se apresentaram pessimistas em relação aos resultados da simulação. Uma razão para esta discrepância se deve à hipótese exponencial do tempo de alocação do canal. Como reconhecido no capítulo 5, a distribuição exponencial que rege este período é de difícil percepção, pois há uma interdependência entre este tempo de alocação e o tempo de espera. A utilização dos modelos analíticos se restringe, desta maneira, a um estudo do pior caso, ou seja, a uma análise pessimista do sistema real, onde o ponto de saturação ocorre sempre antes do mesmo ponto no sistema físico observado.

Para a faixa dada de parâmetros, o "rendez-vous" apresentou-se coerente com o modelo em situações de baixas utilizações do canal. A diferença dos índices no "wormhole" é mais evidente. Como no "rendez-vous", esta diferença é menor para pequenos índices de utilização dos servidores.

Em todas as estratégias, algumas observações em relação aos parâmetros de entrada podem ser realizadas a partir das características dos mecanismos de troca de mensagens e a partir dos modelos analíticos e de simulação. O fluxo de mensagens está associado à frequência de comunicação entre os módulos paralelos de um algoritmo. Um valor crescente deste parâmetro indica um crescimento do tempo de comunicação. Esta variação se deve ao fato de existir um número maior de mensagens na rede, conseqüentemente mais espera e maior probabilidade de bloqueio. O tamanho da mensagem também é fundamental no desempenho das estratégias. O tempo de alocação da linha é

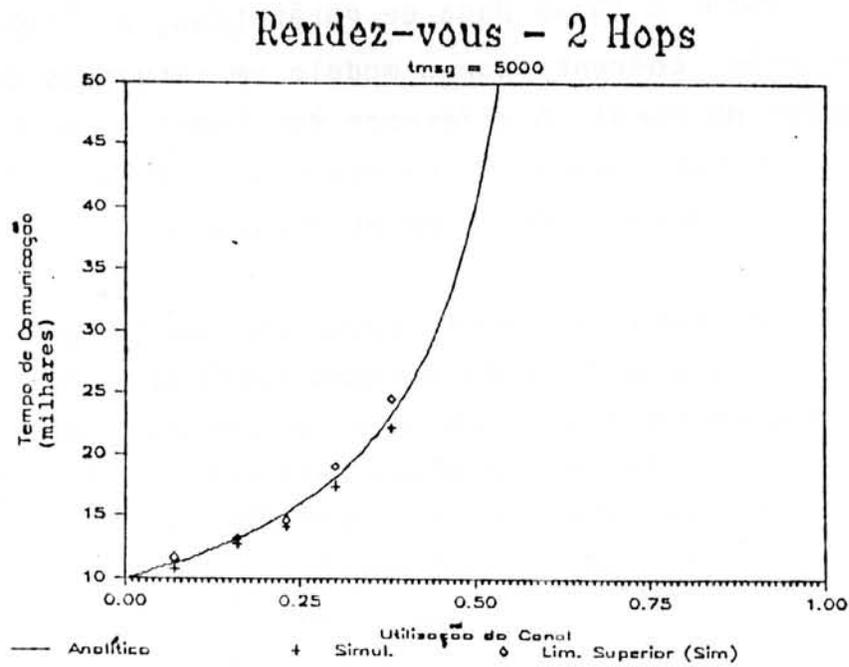
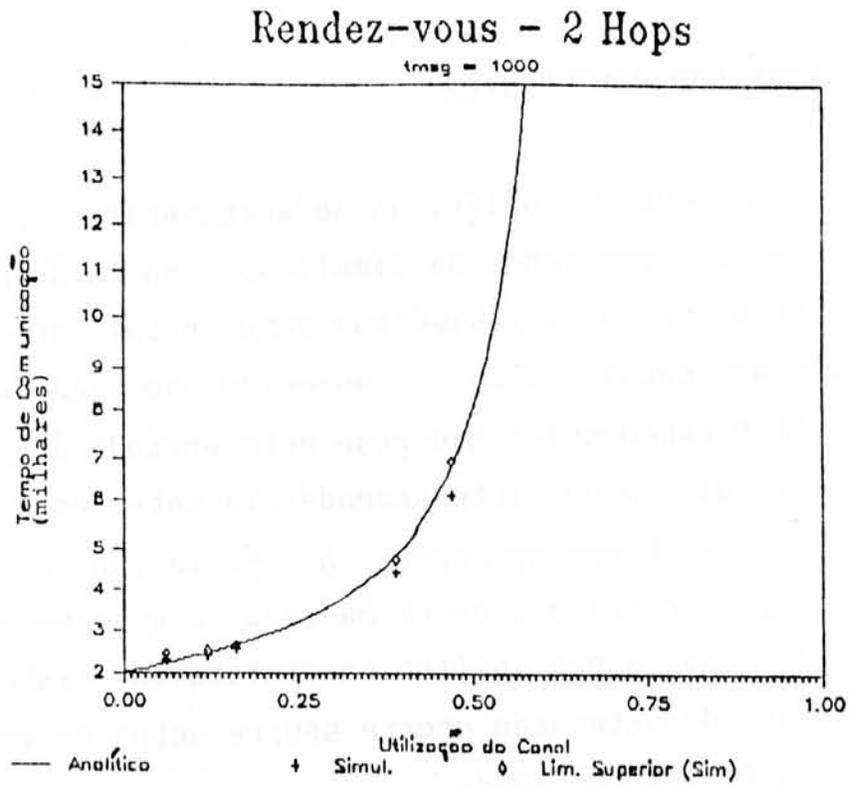


Figura 7.4 - Validação (modelo analítico x modelo em GPSS)
(continua)

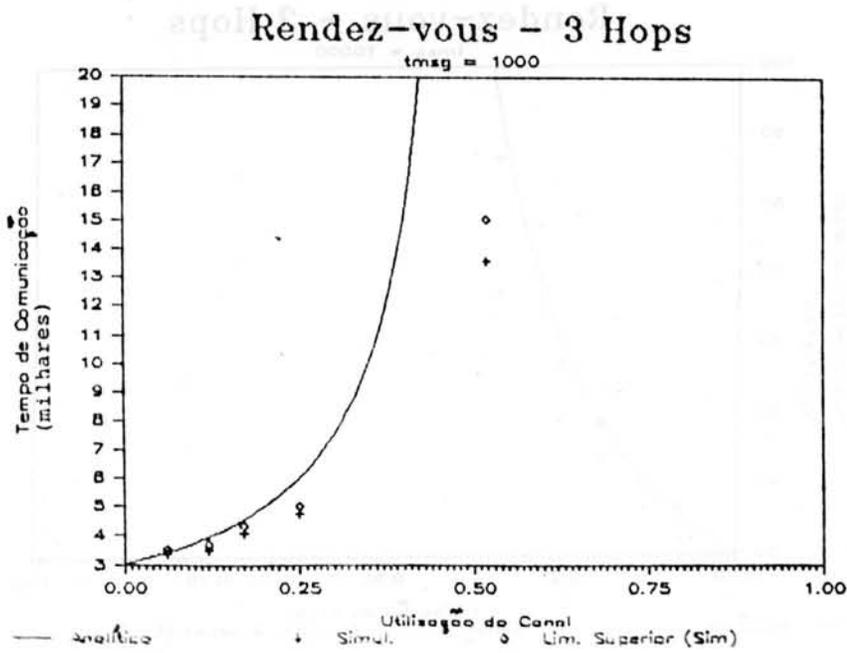
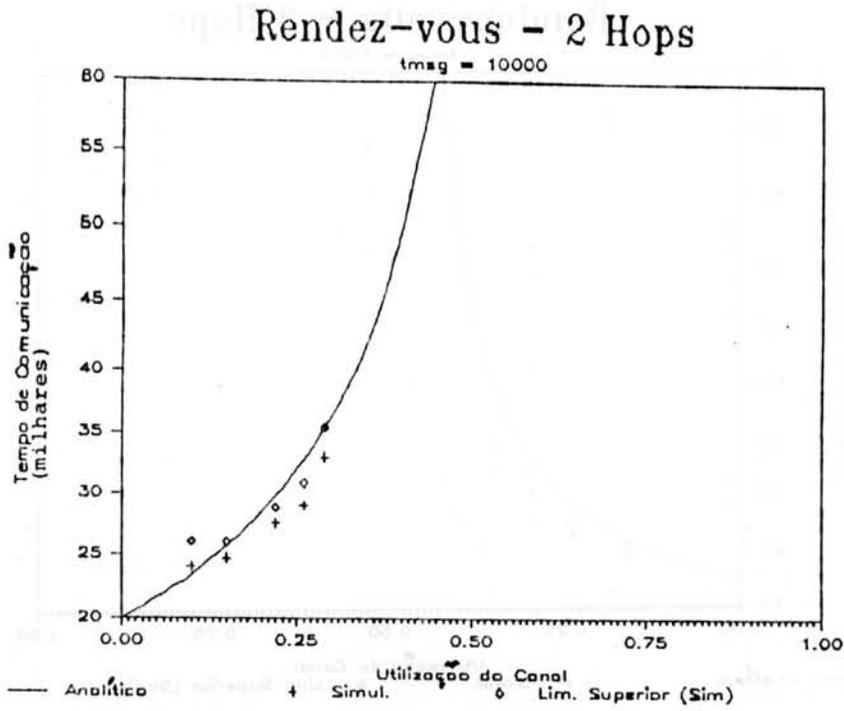


Figura 7.4 - Validação (modelo analítico x modelo em GPSS)
(continuação)

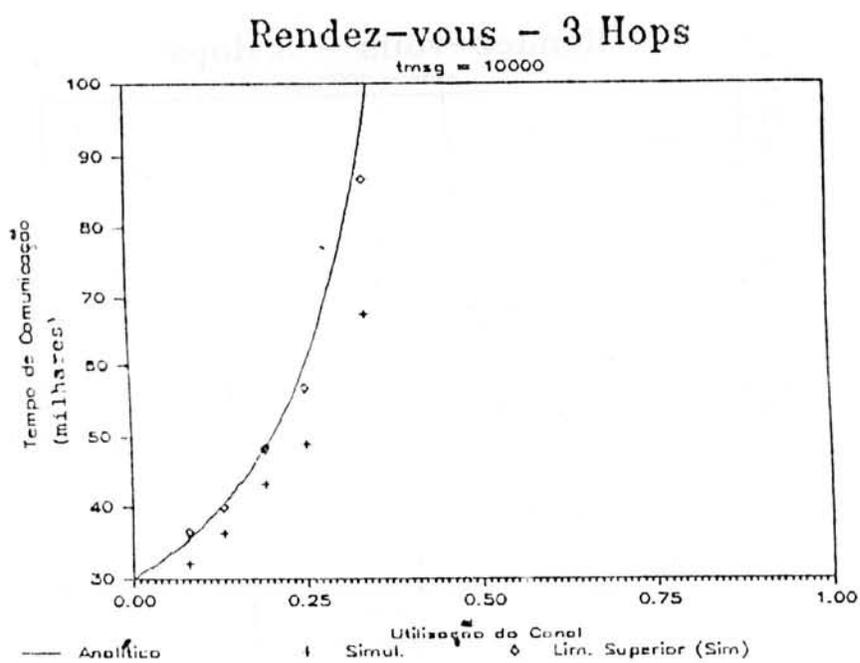
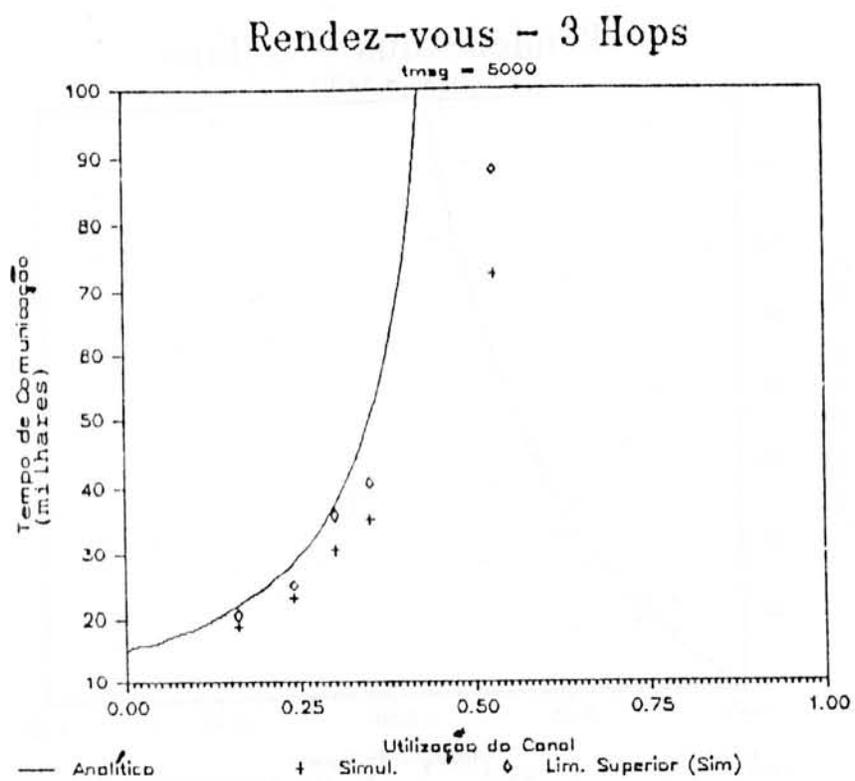


Figura 7.4 - Validação (modelo analítico x modelo em GPSS)
(continuação)

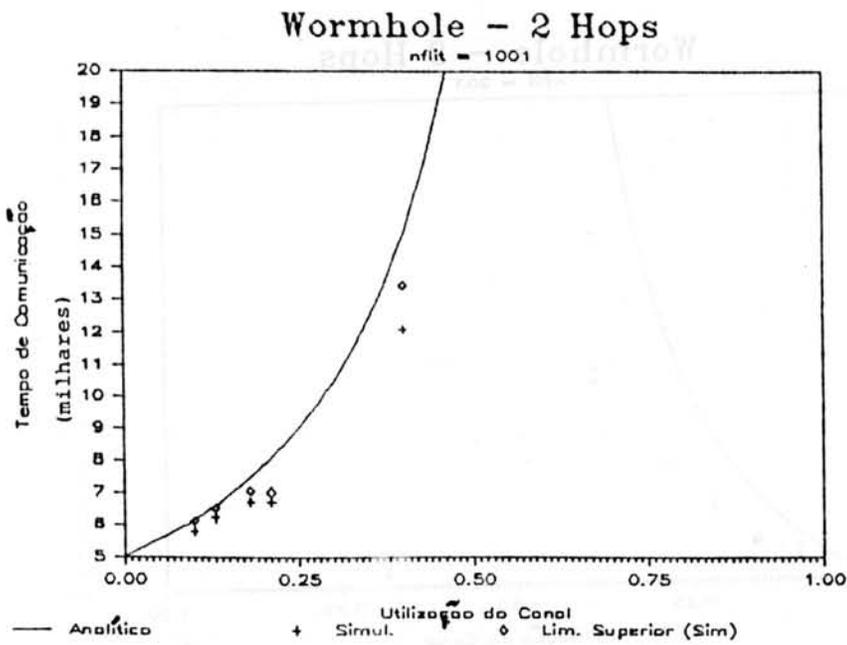
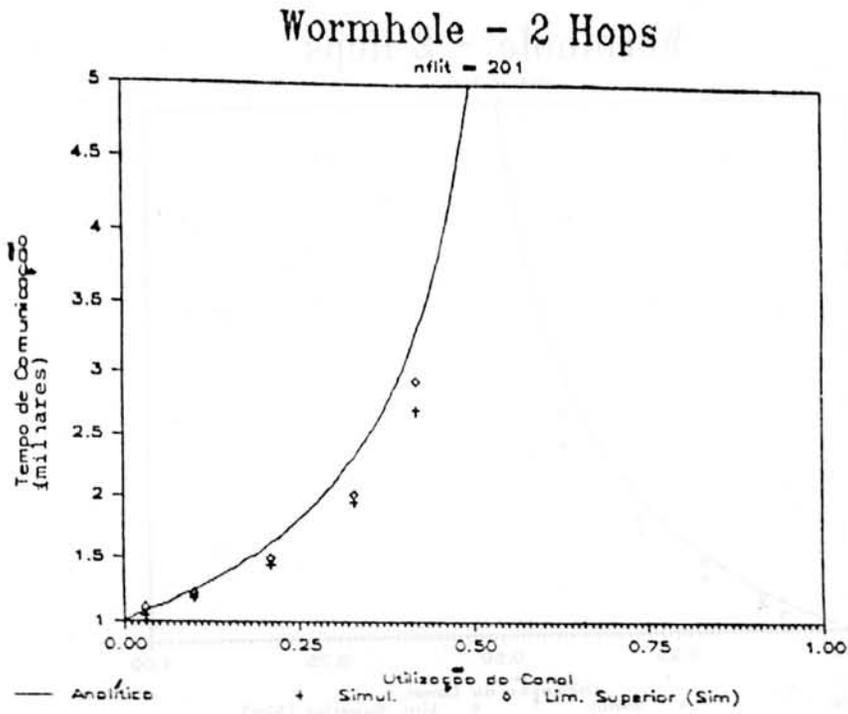


Figura 7.4 - Validação (modelo analítico x modelo em GPSS)
(continuação)

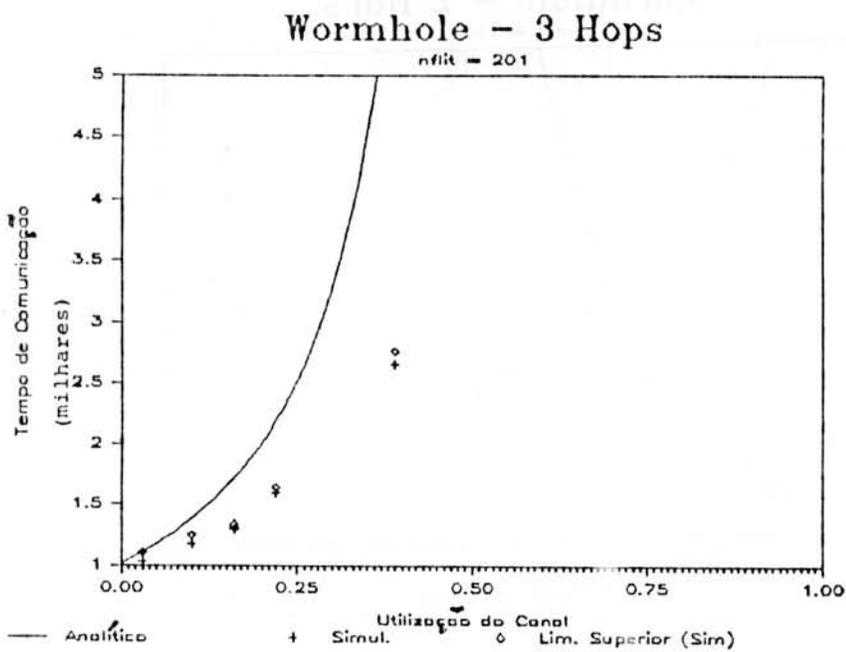
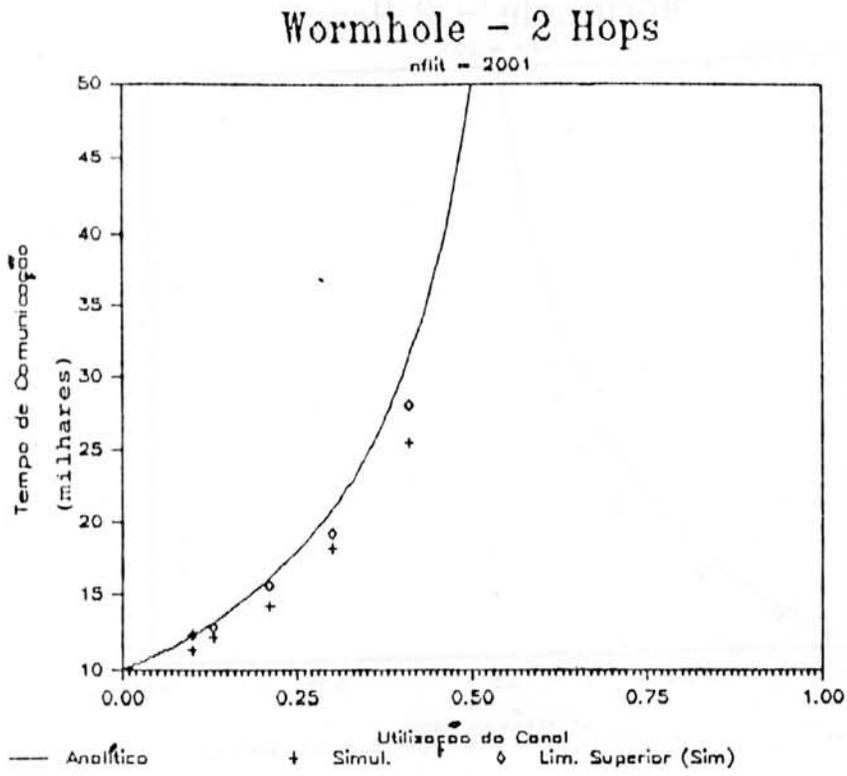


Figura 7.4 - Validação (modelo analítico x modelo em GPSS)
(continuação)

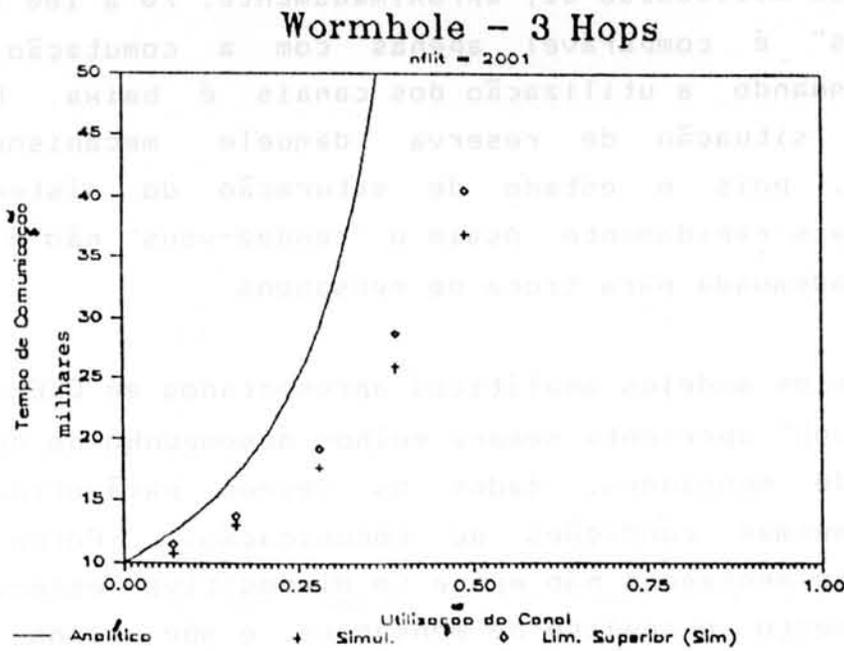
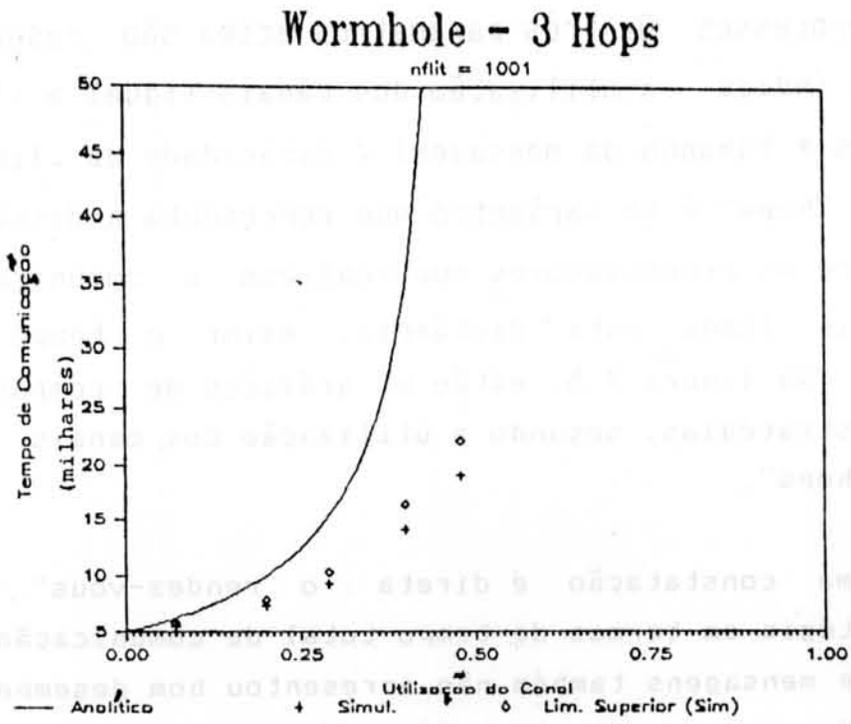


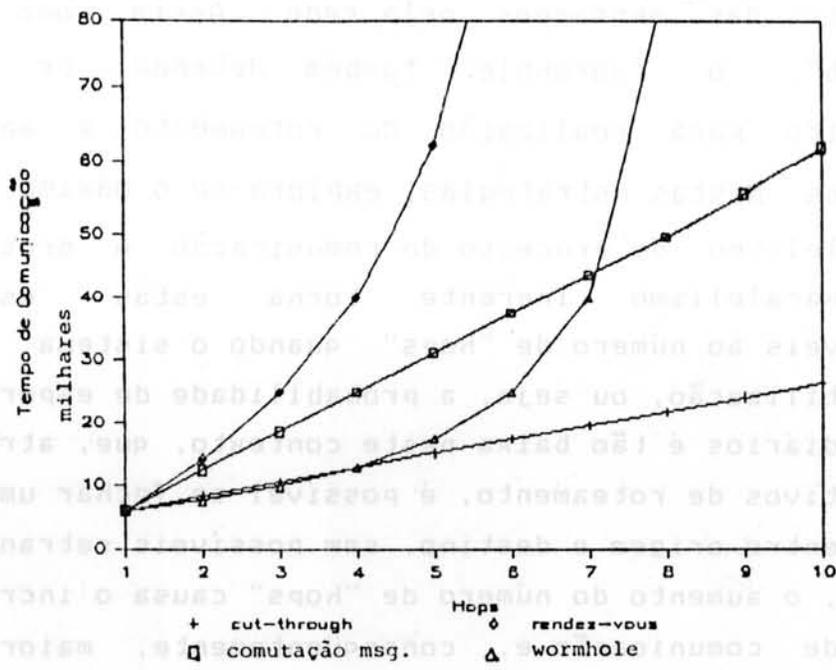
Figura 7.4 - Validação (modelo analítico x modelo em GPSS)
(continuação)

função deste parâmetro : quanto maior a mensagem , maior o tempo de comunicação. A capacidade da linha apresenta, ao contrário do tamanho da mensagem, uma relação inversa : quanto mais rápida a linha, menor o tempo de comunicação entre os processos. Os três parâmetros acima são resumidos em um único índice : a utilização dos canais (igual a (fluxo de mensagens * tamanho da mensagem) / capacidade da linha). O número de "hops" é um parâmetro que representa a distância física entre os processadores que realizam a comunicação. Quanto mais longa esta distância, maior o tempo de comunicação. Na figura 7.5, estão os gráficos de comparação entre as estratégias, segundo a utilização dos canais e o número de "hops".

Uma constatação é direta : o "rendez-vous" é a pior estratégia em termos de tempo total de comunicação. A comutação de mensagens também não apresentou bom desempenho, se comparada com o "wormhole" e o "cut-through" (exceto em ambientes com utilização de, aproximadamente, 70 a 100 %). O "rendez-vous" é comparável apenas com a comutação de mensagens quando a utilização dos canais é baixa. Mesmo assim, a situação de reserva daquele mecanismo é indesejável, pois o estado de saturação do sistema é alcançado mais rapidamente. Assim o "rendez-vous" não é uma estratégia adequada para troca de mensagens.

Pelos modelos analíticos apresentados em [KER 79], o "cut-through" apresenta sempre melhor desempenho do que a comutação de mensagens, dados os mesmos parâmetros de entrada (mesmas condições de comunicação). Porém, a comutação de mensagens não exige um dispositivo específico para roteamento e análise de mensagens, o que torna esta estratégia de utilização mais geral do que o "cut-through". Estas duas estratégias somente possuem desempenho equivalente à medida que a utilização dos canais aumenta, já que o número de "cut-throughs" é menor.

Estratégias de Comunicação



Estratégias de Comunicação

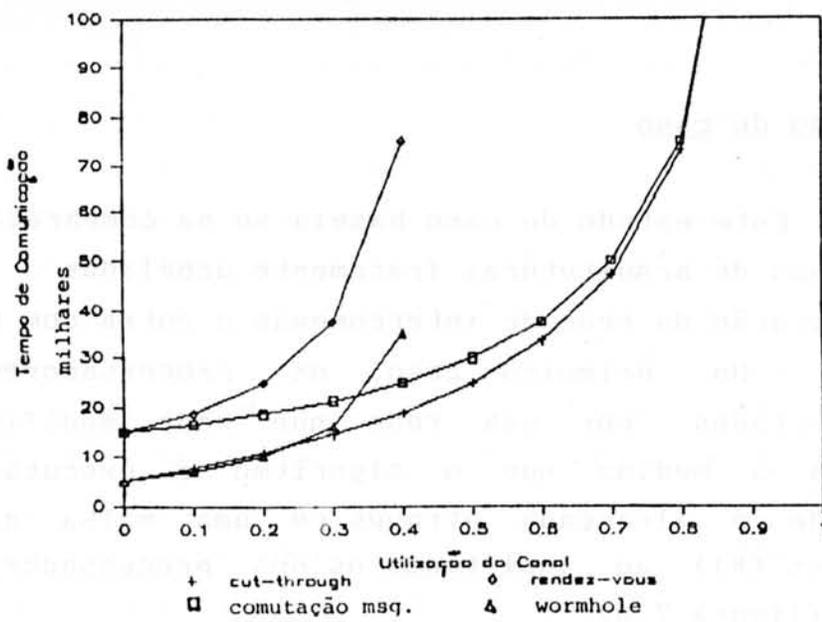


Figura 7.5 - Variação dos parâmetros de comunicação

O "wormhole" é adequado e apresenta melhor desempenho do que o "cut-through" em situações de pouca troca de mensagens. A variação é pequena, mas existe e pode ser importante, de acordo com o tipo da aplicação. Em ambientes de moderada a alta utilização de canais, a característica de bloqueio prejudica sobremaneira o andamento das mensagens pela rede. Assim como o "cut-through", o "wormhole" também depende de hardware específico para realização do roteamento e análise de mensagens. Nestas estratégias, explora-se o máximo possível o paralelismo do processo de comunicação. A grosso modo, este paralelismo inerente torna estas estratégias insensíveis ao número de "hops" quando o sistema apresenta pouca utilização, ou seja, a probabilidade de espera em nós intermediários é tão baixa neste contexto, que, através dos dispositivos de roteamento, é possível se fechar um circuito direto entre origem e destino, sem possíveis retransmissões. Todavia, o aumento do número de "hops" causa o incremento do tempo de comunicação e, conseqüentemente, maior espera. Assim, o "wormhole" torna-se inadequado para comunicações entre processadores distantes entre si.

7.4 Estudo de caso

Este estudo de caso baseia-se na comparação entre dois tipos de arquiteturas fracamente acopladas: uma com reconfiguração da rede de interconexão e outra com topologia estática. No primeiro caso, os processadores estão interconectados por uma rede que pode modificar sua topologia à medida que o algoritmo é executado. Esta capacidade é alcançada através de uma malha de chaves ("switches"(*)) ao qual todos os nós processadores estão ligados (figura 7.6).

Em uma topologia estática, todos os processadores estão interconectados por uma rede estática, imutável. Não

há "switches" (figura 7.7). No multiprocessador reconfigurável, a máquina tenta-se adaptar ao algoritmo. No estático, ocorre o inverso: normalmente, o algoritmo se adapta à arquitetura.

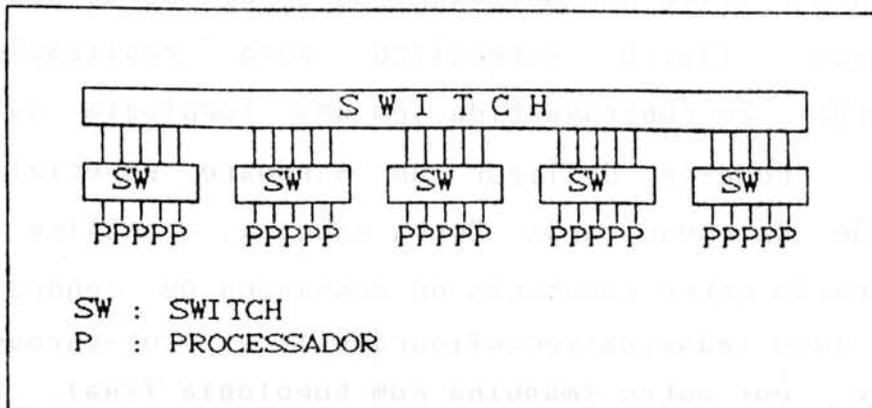


Figura 7.6 - Topologia reconfigurável

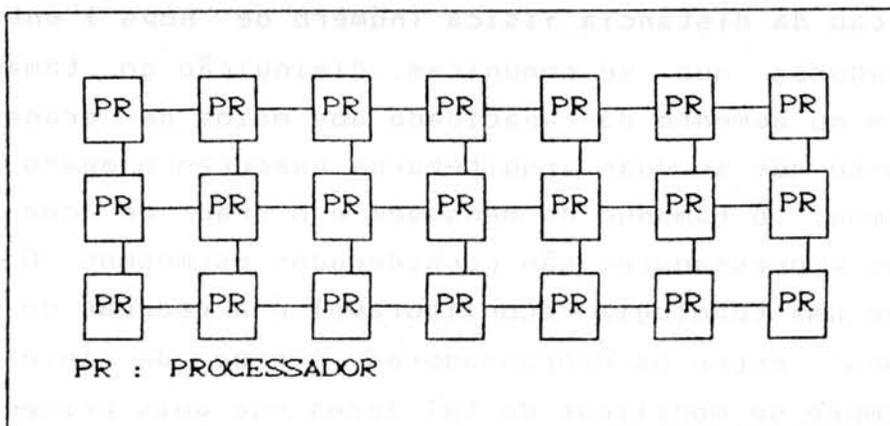


Figura 7.7 - Topologia estática

Um exemplo de arquitetura reconfigurável é o multiprocessador Meganode, construído a partir de

microprocessadores "transputers"(*) [JES 88][TRY 89]. A máquina de Seitz [SEI 89] é um exemplo de máquina com topologia fixa.

Em uma máquina reconfigurável, o roteamento normalmente é feito por software, ou seja, não se utiliza um dispositivo físico específico para realização da comunicação. Em contrapartida, em uma topologia estática, imutável, pode-se utilizar um hardware especial para efetivação da comunicação. Desta maneira, a análise reside na comparação entre comutação de mensagens ou "rendez-vous", por um lado (máquina reconfigurável), e "cut-through" ou "wormhole" por outro (máquina com topologia fixa). Sob as mesmas condições, estas últimas estratégias apresentam um desempenho evidentemente melhor do que aquelas empregadas em ambientes reconfiguráveis.

Existem quatro maneiras de tornar a comunicação mais veloz em uma arquitetura reconfigurável: diminuição do grau de troca de mensagens entre os processadores, diminuição da distância física (número de "hops") entre dois processadores que se comunicam, diminuição do tamanho da mensagem ou aumento da capacidade dos meios de transmissão. Supondo-se que as duas arquiteturas executam o mesmo tipo de algoritmos, o tamanho da mensagem e o grau de comunicação entre os processadores são considerados os mesmos. O que se ganha em uma topologia reconfigurável é a redução do número de "hops" entre os processadores. A rede de interconexão pode sempre se modificar de tal forma que dois processadores (que desejam se comunicar) fiquem próximos. Porém, esta vantagem ocorre às custas de uma baixa velocidade de transmissão dos canais, pois as mensagens passam por redes de chaves. Além disto, o tempo de reconfiguração pode ser considerável. Nas topologias fixas, por outro lado, o número de "hops" que a mensagem deve percorrer é maior, mas a capacidade de escoamento de mensagens pelo canal é também maior, pois os processadores estão interconectados por

ligações diretas, ponto-a-ponto.

Conseqüentemente, a única maneira de haver comunicação eficiente em uma máquina com reconfigurabilidade é a redução do número de "hops" para um, já que o "wormhole" e o "cut-through" são estratégias que apresentam pouca sensibilidade em relação a este parâmetro. Esta situação é difícil de ser alcançada, haja vista a quantidade de diferentes algoritmos que podem executar em um multiprocessador. A reconfigurabilidade da rede de interconexão seria freqüente e, mesmo que a distância entre processadores se reduzisse a um, o tempo gasto e a complexidade seriam altos.

As topologias estáticas, sob o aspecto comunicação, têm apresentado melhor desempenho. A inflexibilidade de sua topologia é compensada por roteamento e canais de comunicação mais rápidos. O roteamento por hardware é, sem dúvida, mais veloz que o roteamento por software. As arquiteturas reconfiguráveis flexíveis são ineficientes porque o "overhead" extra para implementação desta reconfiguração não é desprezível. A solução ideal para este contraste é o desenvolvimento de máquinas reconfiguráveis com algum hardware específico para comunicação ou máquinas estáticas com alguma possibilidade de reconfiguração dinâmica.

The first part of the report deals with the general situation of the country and the progress of the war. It is followed by a detailed account of the military operations in the West, and then a chapter on the situation in the East. The report concludes with a summary of the results of the war and a forecast for the future.

The second part of the report deals with the economic situation of the country and the progress of the war. It is followed by a detailed account of the military operations in the West, and then a chapter on the situation in the East. The report concludes with a summary of the results of the war and a forecast for the future.

8 CONCLUSÃO

O problema da comunicação é fundamental no que concerne desempenho de multiprocessadores fracamente acoplados. A limitação de recursos (processadores e linhas de transmissão) gera uma disputa destes mesmos recursos pelas mensagens.

Um esquema adequado de comunicação deve ser alcançado. As condições de carga da máquina e as próprias características do algoritmo são importantes e nenhum esquema pode ser considerado superior para todos os casos. Neste trabalho, foram analisadas quatro estratégias: comutação de mensagens, semelhante à comutação de pacotes em redes de computadores; "virtual cut-through", uma extensão da comutação de mensagens; "rendez-vous", baseado no protocolo de comunicação da linguagem Occam e "wormhole", idealizada recentemente, baseada em uma transmissão concorrente de segmentos de mensagem.

Na faixa de parâmetros analisada, o "cut-through" apresentou melhor desempenho na maioria dos casos, pois o grau de ocupação ("holding") da linha é menor, em relação aos outros casos. Porém, as mensagens no "cut-through" são transmitidas de maneira seqüencial. O "wormhole" apresenta características de reserva (os canais entre o primeiro "flit" e o último devem estar reservados, pois a maior parte dos "flits" não contém informações de roteamento). Todavia, vários "flits" são transmitidos ao mesmo tempo ao longo de todo o percurso da comunicação. Este paralelismo é benéfico e, através da análise, existem casos onde o "wormhole" apresenta desempenho superior (baixa utilização dos canais). Este ganho não inviabiliza o "wormhole" como esquema de comunicação.

Já o "rendez-vous" e a comutação de mensagens apresentaram pior desempenho em relação às anteriores, sob

as mesmas condições de comunicação. Estas estratégias podem ser adotadas, principalmente em arquiteturas que não dispõem de equipamento específico para roteamento. Por exemplo, na comparação entre uma topologia estática e uma dinâmica, o "rendez-vous" pode ser mais atrativo, se as comunicações são mais curtas, mesmo com uma perda da capacidade de transmissão das linhas, devido à hierarquia de chaves que a mensagem eventualmente pode passar. Ainda assim, o seu desempenho é comprovadamente menor em relação às outras estratégias. A comutação de mensagens só é comparável ao "cut-through" em casos de alta utilização de canais.

Toda a análise desenvolvida neste trabalho foi feita sobre modelos bastante genéricos e dentro de uma faixa limitada de parâmetros. Os resultados obtidos dos mesmos podem discordar da realidade. Porém, a diferença esperada corresponde ao compromisso entre precisão e simplicidade do modelo. A quantificação da superioridade desta ou daquela estratégia não faz parte dos objetivos pretendidos. Se há necessidade de resultados numéricos, uma sugestão a seguir é o refinamento dos modelos. Porém, a modelagem de situações de bloqueio e de reserva é complicada e a precisão pode ser alcançada às custas de uma grande complexidade dos modelos. Na figura 8.1, encontra-se um quadro resumo das quatro estratégias abordadas e uma síntese das suas principais características.

Alguns aspectos não abordados constituem fonte de pesquisa : a influência dos algoritmos de roteamento, considerações sobre os tempos de processamento e de propagação (capítulo 4), tolerância a falhas, determinação da topologia ótima. Sugere-se também a comparação dos modelos analíticos e de simulação com índices fornecidos diretamente por um multiprocessador.

Quanto à aplicabilidade do trabalho, a questão é transportada a um nível mais alto e o questionamento deve

ser enfatizado sobre a importância da avaliação de desempenho de sistemas computacionais. O projeto de qualquer arquitetura ou o desenvolvimento de qualquer programa deve sempre levar em consideração aspectos de eficiência e principalmente de custo, o parâmetro mais importante para a viabilidade de um projeto. No momento em que se aceita o estudo da avaliação de sistemas computacionais, determina-se também a importância da análise do presente trabalho.

	"rendez-vous"	com. mensagem	"wormhole"	"cut-through"
1	reserva canais	não reserva canais	reserva canais	não reserva canais
2	canais ociosos	canais : plena utilização	canais ociosos	canais : plena utilização
3	adequado para baixas utilizações	adequado para qualquer utilização	adequado para baixas utilizações	adequado para qualquer utilização
4	solicitação ack, mensagem *	mensagem *	"flits" **	mensagem *
5	por software	por software	por hardware	por hardware

* cabeçalho + dados

** "flits" de cabeçalho + "flits" de dados

Legenda da tabela (características de cada estratégia):

- 1 - reserva (bloqueio) dos canais de comunicação
- 2 - nível de utilização dos canais de comunicação
- 3 - contexto adequado para a estratégia
- 4 - tipos de mensagens necessários para a comunicação entre os processadores
- 5 - roteamento

Figura 8.1 - Síntese das estratégias de comunicação
(continua)

parâmetro	tempo de comunicação
número de "hops"	aumenta
tamanho da mensagem (*)	aumenta
capacidade da linha (*)	diminui
fluxo de mensagens (lambda) (*)	aumenta

(*)
$$\text{utilização da linha} = \frac{\text{tamanho da mensagem} * \text{fluxo de mensagens}}{\text{capacidade da linha}}$$

Obs. a tabela indica a variação do tempo de comunicação em função do incremento do parâmetro correspondente

Figura B.1 - Síntese das estratégias de comunicação
(continuação)

A N E X O S

THE END

ANEXO 1

TEORIA DE FILAS - O MODELO M/M/1 [KLE 75]

Um sistema de enfileiramento ("queueing system") caracteriza-se por um recurso de capacidade finita sobre o qual demandas são impostas. O recurso é denominado servidor e as demandas, clientes. Quando o servidor está ocupado com um cliente, os demais esperam.

Para especificar completamente um sistema de filas, é necessário identificar (figura A.1) :

- o processo de chegada descrito por uma distribuição de probabilidade dos intervalos de tempos entre chegadas de clientes
- o processo de serviço descrito por uma distribuição de probabilidade do tempo de serviço do servidor. Este tempo de serviço corresponde à quantidade de tempo que o cliente passa no servidor. As distribuições mais comuns, tanto para o processo de serviço, como para o processo de chegada, são a exponencial (ou markoviana), de Erlang, hiperexponencial, determinística e geral
- a capacidade da área dedicada aos clientes em espera, ou seja, o número de clientes que cabem na fila
- o número de estações de serviço
- as classes de clientes, onde cada classe pode ter sua própria distribuição de chegada e de serviço
- a disciplina da fila, isto é, o critério de atendimento dos clientes : FIFO ("first-in first-out" - o primeiro que chega é o primeiro a ser atendido), LIFO ("last-in first-out" - o último a chegar é o primeiro a ser atendido), aleatório, etc.

- a prioridade de clientes, seja entre classes, seja dentro de uma mesma classe
- o grau de disponibilidade do servidor, se o mesmo é alocado ocasionalmente a outras tarefas, como, por exemplo, sua própria manutenção
- o comportamento dos clientes, que podem abandonar a fila, "subornar outros clientes a fim de obter uma melhor posição" (sic [KLE 75]), mudar de fila, ou mesmo recusar a entrar no sistema, se a mesma estiver longa
- o tamanho da população de clientes.

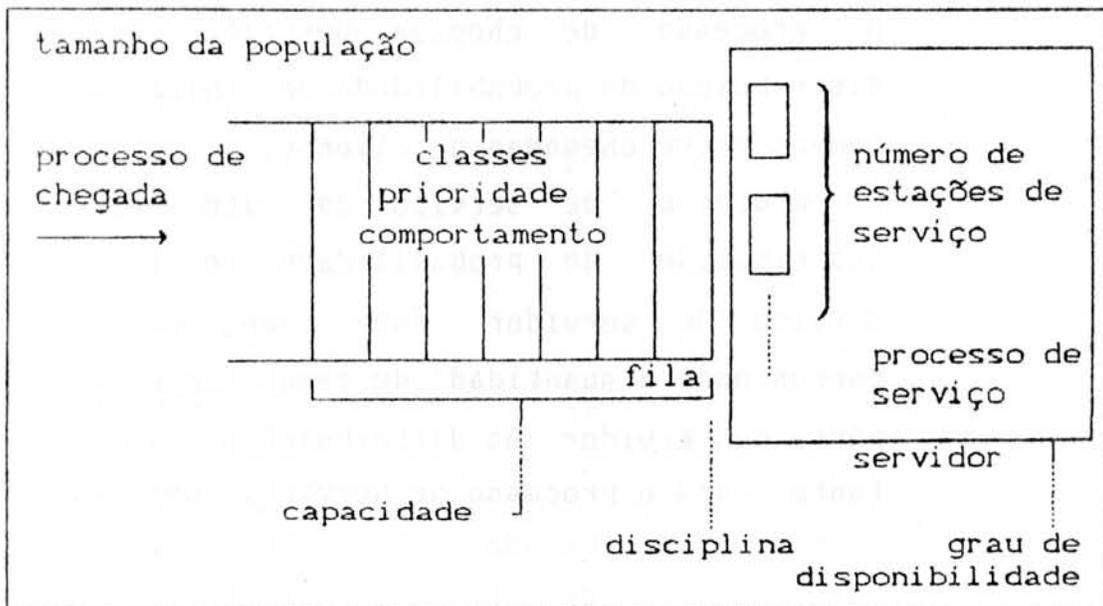


Figura A.1 - Representação gráfica de uma fila

A notação comumente utilizada é :

$A/B/m/K/M$

onde A representa a distribuição do tempo entre chegadas, B a distribuição do tempo de serviço, m o número de servidores, K a capacidade da fila e M o tamanho da população de clientes. Os três primeiros componentes (A, B, m) são obrigatórios. Os demais (K e M), quando ausentes,

indicam um valor infinito. As demais especificações do sistema de fila são indicadas de forma textual.

O número de diferentes sistemas que podem ser construídos a partir da combinação dos aspectos acima é infinito. Para compreender um pouco como deduzir e aplicar os resultados da teoria de filas, será analisado o modelo mais simples : o M/M/1. A letra M indica que as distribuições do tempo entre chegadas de clientes e de serviço são exponenciais (ou markovianas). Há somente um servidor e a capacidade da fila, assim como o tamanho da população, é ilimitada. Não há classe de clientes, nem prioridade. O primeiro cliente que chega é o primeiro a ser atendido. O servidor está alocado somente para servir aos clientes, que se juntam à fila, são atendidos e partem do sistema após receber o serviço.

Neste anexo, será demonstrado de forma intuitiva como calcular o tempo médio gasto pelos clientes no sistema, quando o mesmo se encontra no equilíbrio. Este resultado é possível de ser calculado graças à lei de conservação do fluxo. Segundo esta lei, todo sistema em equilíbrio apresenta igual fluxo de entrada e de saída. Suponha um recipiente furado com um litro de água no seu interior, alimentado constantemente por um jato de água, proveniente de uma torneira. Se a taxa de escoamento do recipiente (saída) é a mesma da torneira (entrada), a quantidade de água permanecerá a mesma.

Em uma fila, a lei é aplicada para cada estado do sistema : no equilíbrio, o fluxo de transição para um determinado estado é o mesmo que o fluxo de transição a partir daquele estado.

Uma fila é considerada um sistema de nascimento e morte. A chegada de clientes é o processo de "nascimento" e o serviço, o processo de "morte". O estado de uma fila com

tempos entre chegadas e de serviço exponenciais pode ser completamente definido pelo número de clientes no sistema (em serviço mais em espera). O intervalo de tempo para a próxima chegada e o intervalo de tempo até o fim do serviço do cliente corrente no servidor não interessam, pois as distribuições são markovianas : elas têm a propriedade de não possuírem memória ("memoryless"), ou seja, "a probabilidade de ocorrência de qualquer estado futuro, dados os estados presentes e passados, é independente dos eventos passados e depende somente do estado presente do processo" [KOB 78].

O sistema M/M/1 pode ser representado por um diagrama de estados. Cada círculo representa um estado, ou seja, o número de clientes no sistema. Cada seta indica qual a transição permitida e qual a taxa de sua ocorrência. No caso, λ corresponde à taxa de chegada de clientes e μ , a taxa de processamento de clientes pelo servidor (figura A.2).

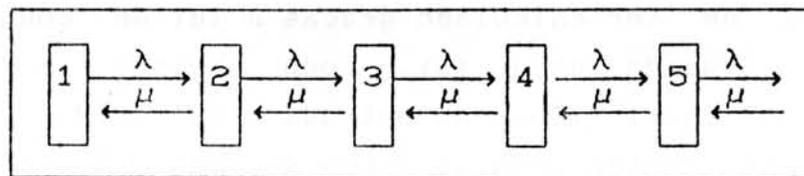


Figura A.2 - Diagrama de transição de estados para M/M/1

Se se observa um estado arbitrário, o fluxo de entrada do mesmo corresponde à soma das taxas de transição para aquele estado, ponderada pelas probabilidades de ocorrência dos estados vizinhos, onde p_k representa a probabilidade do sistema estar no estado k .

$$\text{taxa de entrada} = \lambda p_{k-1} + \mu p_{k+1} \quad (12)$$

Analogamente, o fluxo de saída corresponde à soma das taxas de transição, multiplicada pela probabilidade de ocorrência do estado em questão. A ponderação é obrigatória, pois o sistema não é determinístico, ou seja, no equilíbrio, os seus estados respeitam uma variável aleatória.

$$\text{taxa de saída} = (\lambda + \mu)p_k \quad (13)$$

No equilíbrio, as duas expressões acima são equivalentes.

$$\lambda p_{k-1} + \mu p_{k+1} = (\lambda + \mu)p_k \quad (14)$$

A equação de equilíbrio para o estado 0 é dada pela equação abaixo.

$$p_1 \mu = \lambda p_0 \quad (15)$$

Para o estado 1, tem-se :

$$p_1 = \frac{\lambda}{\mu} p_0 \quad (16)$$

Com a aplicação da fórmula (15), tem-se :

$$0 = -(\lambda + \mu)p_1 + \lambda p_0 + \mu p_2$$

$$0 = -(\lambda + \mu) \frac{\lambda}{\mu} p_0 + \lambda p_0 + \mu p_2 \quad (17)$$

$$0 = -\frac{\lambda\lambda}{\mu} p_0 - \lambda p_0 + \lambda p_0 + \mu p_2$$

Como resultado, obtém-se a equação abaixo que corresponde à probabilidade de ocorrência do estado 2.

$$p_2 = \frac{\lambda\lambda}{\mu\mu} p_0 \quad (18)$$

Através de manipulação algébrica e indução, pode-se chegar a expressão geral para p_k .

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda}{\mu} \quad (19)$$

Aplicando-se o somatório em ambos os lados da equação acima, obtém-se:

$$1 = p_0 \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda}{\mu} \quad (20)$$

A condição de estabilidade do sistema é que a taxa da processamento do servidor seja maior que a taxa de chegada de clientes. Caso contrário, as filas cresceriam indefinidamente e o servidor ficaria sobrecarregado de serviço. Se $\lambda < \mu$, pode-se aplicar a expressão da soma de seqüências infinitas (P_0 - probabilidade do sistema estar vazio).

$$1 = \frac{1}{1 - (\lambda/\mu)} P_0$$

$$P_0 = 1 - \frac{\lambda}{\mu} \quad (21)$$

E a expressão para P_k pode ser reescrita em função do resultado anterior.

$$P_k = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^k \quad (22)$$

A razão entre λ e μ é denominada utilização do sistema e é representado por ρ .

$$\rho = \frac{\lambda}{\mu} \quad (23)$$

Desta forma, o número médio de clientes (N) é determinado pela expressão da média :

$$N = \sum_{k=0}^{\infty} k p_k \quad (24)$$

Com manipulação algébrica, chega-se à expressão do número médio de clientes no sistema :

$$N = \frac{\rho}{1 - \rho} \quad (25)$$

Quando um cliente chega ao sistema, ele encontra, em média, N clientes na fila. O tempo que ele vai gastar no sistema (T) é N vezes o tempo de serviço mais o tempo de serviço dele. Assim, o tempo médio de passagem de clientes pelo sistema é dado por :

$$\begin{aligned} T &= N * TS + TS \\ T &= \frac{\rho}{\mu(1 - \rho)} + \frac{1}{\mu} \\ T &= \frac{\rho + 1 - \rho}{\mu(1 - \rho)} \\ T &= \frac{1/\mu}{1 - \rho} \end{aligned} \quad (26)$$

O tempo médio de estadia é composto do tempo médio de espera (W) mais o tempo médio de serviço (S), indicados pelas equações abaixo :

$$W = \frac{\rho/\mu}{1 - \rho}$$
$$S = \frac{1}{\mu}$$
(27)

Outras medidas de eficiência e de desempenho podem ser calculadas, tais como comprimento de um período ocioso, comprimento do período de ocupação e a quantidade restante (em unidades de tempo) de trabalho a ser realizado. Embora seja desejável determinar uma descrição probabilística completa, a especificação dos primeiros momentos (média e variância) normalmente é suficiente, como no caso acima, onde se calculou somente a média de clientes no sistema e o tempo médio de passagem pelo sistema.

1874

1875

1876

1877

1878

1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900

ANEXO 2

GPSS [JAC 80]

GPSS ("General Purpose Simulator System" - Sistema Simulador de Propósito Geral) simula sistemas representados por uma série de blocos interconectados, pelos quais transações (exemplos : clientes, mensagens, partes de máquinas, veículos) se movimentam. Alguns exemplos típicos de blocos :

1. GENERATE : cria transações
2. QUEUE : cria uma fila de transações e mantém certas estatísticas de enfileiramento
3. TABULATE : tabula informações referentes à simulação
4. TERMINATE : elimina as transações do sistema.

A seqüência de eventos do sistema é simulada pelo movimento de transações de bloco para bloco. Pode haver muitas transações diferentes no modelo, mas em um dado instante cada uma delas está localizada em um bloco, que pode conter e atuar sobre uma ou mais transações simultaneamente. Transações são entidades temporárias que são geradas, movidas através dos blocos e removidas pelo programa em GPSS. Algumas transações são permanentes e permanecem presentes no sistema durante todo o período da execução da simulação. Blocos que podem manipular somente um item por vez são chamados facilidades. Um exemplo de facilidade é uma máquina que realiza alguma tarefa sobre um objeto (transação). Uma área de armazenamento ("storage") representa um outro tipo de bloco que pode ser utilizado por uma ou mais transações simultaneamente. Um exemplo de área de armazenamento é um "buffer" de mensagens.

GPSS modela o avanço do tempo através do bloco ADVANCE. Quando a transação entra neste bloco, um tempo de

ação é computado e adicionado ao tempo corrente, a fim de produzir, para uma transação, o instante de saída do bloco. Quando o relógio do simulador indica este tempo de saída, a transação se move, se possível, para o próximo bloco do diagrama. Transações podem possuir certos atributos que podem ser utilizados para realizar decisões lógicas dentro do bloco. O programa é também capaz de coletar dados estatísticos sobre o processo simulado. Por exemplo, o número corrente de transações em uma área de armazenamento ou o comprimento de uma fila.

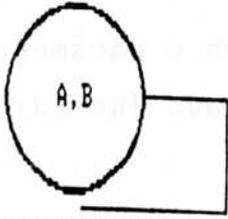
Cálculos matemáticos simples podem ser realizados através de comandos sobre variáveis. Várias distribuições numéricas de probabilidade podem ser geradas em GPSS. A saída do programa pode ser produzida em uma forma padrão ou é possível criar um relatório segundo especificações do usuário (estas facilidades dependem da versão do simulador). Erros de programação podem ser detectados pelo montador ou durante a execução da simulação. GPSS é uma linguagem adequada para modelagem de problemas de enfileiramento.

Segundo [JAC 80], GPSS apresenta as seguintes características :

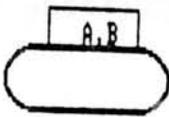
1. restrita a problemas de enfileiramento
2. não é possível estender a linguagem
3. fácil de aprender e usar, proporciona um ambiente conceitual simples
4. boas facilidades de depuração, incluindo diagnósticos compreensivos
5. entrada/saída inflexíveis : estatísticas sempre geradas
6. facilidades de computação inadequadas
7. eficiência de execução pobre pois o programa em GPSS é sempre interpretado.

A simulação conduzida neste trabalho utilizou uma versão para microcomputador PC-IBM compatível [GPS 83].

Os principais blocos utilizados foram :



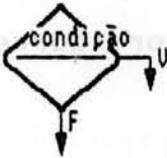
GENERATE : gera mensagens a cada A unidades de tempo (média), segundo uma distribuição definida por B



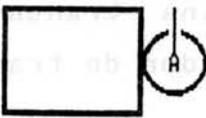
ASSIGN : coloca no parâmetro A da transação o valor correspondente a B



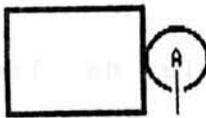
TRANSFER : transfere incondicionalmente



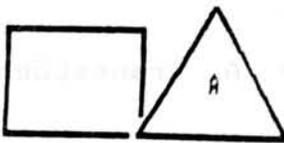
TEST : transfere condicionalmente



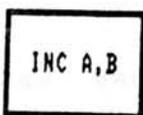
QUEUE : enfileira na fila A



DEPART : sai da fila A



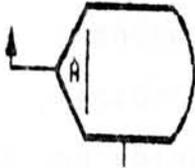
SEIZE : obtém facilidade A



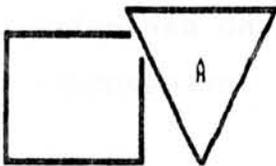
INCREMENT : incrementa parâmetro A de B unidades



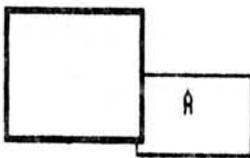
ADVANCE : avança o relógio da simulação de A unidades de tempo



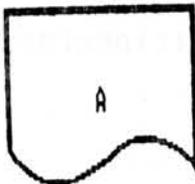
LOOP : diminui de um o parâmetro A e desvia se o resultado for diferente de zero



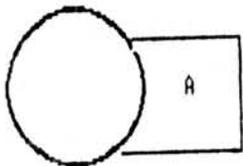
RELEASE : libera a facilidade A



TABULATE : tabula valor específico na tabela A



PRINT : imprime valor corrente do atributo A



TERMINATE : elimina transação e diminui de A o contador de transações

Os principais comandos de controle da linguagem são :

START A

Inicia a simulação até que A transações sejam processadas.

RESET

Zera estatísticas da simulação, mas mantém no sistema as transações correntes.

SIMULATE

Indica que o modelo não deve ser somente montado, como também efetivamente simulado.

REALO XAC,A

O número máximo de transações simultâneas é dado por A.

A EQU B

O rótulo A é dado ao endereço B.

A FUNCTION B,C $x_1, y_1/x_2, y_2/...../x_c, y_c$

é definida a distribuição de probabilidade A, definida pela variável randômica B e com C intervalos de determinação da imagem. A função é especificada pelos pontos x_i, y_i .

A FVARIABLE (expressão)

A expressão é avaliada a cada referência de A. Todas as variáveis são consideradas como de ponto flutuante. O resultado final é truncado nas casas decimais.

A VARIABLE (expressão)

A expressão é avaliada a cada referência de A. Antes da avaliação, todas as variáveis de ponto flutuante são truncadas nas casas decimais.

A TABLE B,C,D,E

A tabela A é definida. São tabulados os parâmetros B por faixas (por exemplo, se B é M\$1, os tempos de trânsito das transações serão tabulados). C corresponde ao limite superior do intervalo mais baixo. D é o comprimento de uma faixa e E corresponde ao número de faixas.

INITIAL A,B

Inicializa a variável A com o valor de B.

END

Indica fim do programa da simulação.

Nos anexos 3 e 4 são apresentados exemplos documentados dos programas utilizados neste trabalho : simulação do "rendez-vous" em um anel e em um vetor e simulação do "wormhole" também em um anel e em um vetor. O número de processadores é 10.

ANEXO 3

PROGRAMAS EM GPSS DO MODELO DE SIMULAÇÃO DO "RENDEZ-VOUS"

```

*
*   Nome      : RENDVETO.GPS
*   Estrategia : "rendez-vous"
*   Topologia  : vetor (10 processadores)
*   Autor     : Joao Paulo F. W. Kitajima
*
*
*   Alocacao de espaco para transacoes
*
*   REALD     XAC,3000
*
*   Rotulo da funcao exponencial
EXPON EQU    1
*
*   Rotulo da tabela de tempos de comunicacao
CTIME EQU    2
*
*   Tempo medio entre geracoes entre mensagens
INTER EQU    3
*
*   Tempo de transmissao da solicitacao
TSOL EQU     4
*
*   Tempo de transmissao do aceite
TACK EQU     5
*
*   Tempo de transmissao da mensagem
TMSG EQU     6
*
*   Numero de "hops"
HOPS EQU     7
*
*   Funcao exponencial
EXPON FUNCTION RN$1,C24

```

0,0/.100,.104/.200,.222/.300,.355/.400,.509
 .500,.690/.600,.915/.700,1.200/.750,1.380
 .800,1.600/.840,1.830/.880,2.120/.900,2.300
 .920,2.520/.940,2.810/.950,2.990/.960,3.200
 .970,3.500/.980,3.900/.990,4.600/.995,5.300
 .998,6.200/.999,7/.9998,8

*

* Escolhe um tamanho variavel de mensagem

TMSG FVARIABLE FN\$EXPON*10000

*

* Define uma tabela de tempos de comunicacao

CTIME TABLE M\$1,0,20000,5

*

* Inicializa tempo entre geracao de mensagens

INITIAL X\$INTER,400000

*

* Inicializa tempo de transmissao da solicitacao

INITIAL X\$TSOL,5

*

* Inicializa tempo de transmissao do ACK

INITIAL X\$TACK,1

*

* Inicializa numero de "hops"

INITIAL X\$HOPS,2

*

SIMULATE

*

GENERATE X\$INTER, FN\$EXPON, , , , 5, F

ASSIGN 1, 1

TRANSFER , HOP

*

GENERATE X\$INTER, FN\$EXPON, , , , 5, F

ASSIGN 1, 2

TRANSFER , HOP

*

GENERATE X\$INTER, FN\$EXPON, , , , 5, F

ASSIGN 1, 3

```

TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 4
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 5
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 6
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 7
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 8
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 9
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 5, F
ASSIGN        1, 10
TRANSFER      ,HOP
*
*
*
HOP ASSIGN     3, X$HOPS
ASSIGN        2, P$1
ASSIGN        4, P$3
*
* Transmite solicitacao

```

```

*
BACK  QUEUE      *1
      SEIZE      *1
      DEPART     *1
      INCREMENT  1,1
      ADVANCE    X$TSOL
      LOOP       3,BACK

*
      ASSIGN     3,P$4

*
*   Transmite ACK
*
BACK2 ADVANCE    X$TACK
      LOOP       3,BACK2

*
      ASSIGN     1,P$2
      ASSIGN     3,P$4
      ASSIGN     5,V$TMSG

*
*   Transmite mensagem
*
BACK3 ADVANCE    *5
      RELEASE    *1
      INCREMENT  1,1
      LOOP       3,BACK3

*
      TABULATE   CTIME
SAI   TERMINATE  0

*
*   Relogio - tempo simulado
*
      GENERATE   10000000

*   TB$2 - tempo de comunicacao - valor da tabela CTIME
*   FR$5 - utilizacao do canal 5
*   AC$1 - valor do relógio do simulador
PRINT  FR$5,TB$2,AC$1
TERMINATE 1

```

*

* Periodo transiente

*

START 1

RESET

*

* Equilibrio

*

START 1

RESET

START 1

END

```

*
*   Nome      : RENDANEL.GPS
*   Estrategia : "rendez-vous"
*   Topologia  : anel (10 processadores)
*   Autor     : Joao Paulo F. W. Kitajima
*
*
*   Alocacao de espaco para transacoes
*
*   REALD    XAC,3000
*
*   Rotulo da funcao exponencial
EXPON EQU      1
*
*   Rotulo da tabela de tempos de comunicacao
CTIME EQU      2
*
*   Tempo medio entre geracoes entre mensagens
INTER EQU      3
*
*   Tempo de transmissao da solicitacao
TSOL  EQU      4
*
*   Tempo de transmissao do aceite
TACK  EQU      5
*
*   Tempo de transmissao da mensagem
TMSG  EQU      6
*
*   Linha corrente de transmissao
LINE  EQU      7
*
*   Numero de "hops"
HOPS  EQU      8
*
*   Funcao exponencial
EXPON FUNCTION  RN$1,C24

```

0,0/.100,.104/.200,.222/.300,.355/.400,.509
 .500,.690/.600,.915/.700,1.200/.750,1.380
 .800,1.600/.840,1.830/.880,2.120/.900,2.300
 .920,2.520/.940,2.810/.950,2.990/.960,3.200
 .970,3.500/.980,3.900/.990,4.600/.995,5.300
 .998,6.200/.999,7/.9998,8

*

*

* Escolhe um tamanho variavel de mensagem

TMSG FVARIABLE FN\$EXPON*10000

*

* Ajusta numero da ligacao em um anel

LINE VARIABLE (P\$1 @ 20) + 1

*

* Define uma tabela de tempos de comunicacao

CTIME TABLE M\$1,0,20000,5

*

* Inicializa tempo entre geracao de mensagens

INITIAL X\$INTER,400000

*

* Inicializa tempo de transmissao da solicitacao

INITIAL X\$TSOL,4

*

* Inicializa tempo de transmissao do ACK

INITIAL X\$TACK,1

*

* Inicializa numero de "hops"

INITIAL X\$HOPS,2

*

SIMULATE

*

GENERATE X\$INTER, FN\$EXPON, , , , 6, F

ASSIGN 1, 1

TRANSFER , HOP

*

GENERATE X\$INTER, FN\$EXPON, , , , 6, F

ASSIGN 1, 2

```

TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 3
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 4
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 5
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 6
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 7
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 8
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 9
TRANSFER      ,HOP
*
GENERATE      X$INTER, FN$EXPON, , , , 6, F
ASSIGN        1, 10
TRANSFER      ,HOP
*
*
*
HOP  ASSIGN    3, X$HOPS

```

```

        ASSIGN      2,P$1
        ASSIGN      4,P$3
*
*   Transmite solicitacao
*
BACK    ASSIGN      6,V$LINE
        QUEUE       *6
        SEIZE       *6
        DEPART      *6
        INCREMENT   1,1
        ADVANCE     X$TSOL
        LOOP        3,BACK
*
        ASSIGN      3,P$4
*
*   Transmite ACK
*
BACK2   ADVANCE     X$TACK
        LOOP        3,BACK2
*
        ASSIGN      1,P$2
        ASSIGN      3,P$4
        ASSIGN      5,V$TMSG
*
*   Transmite mensagem
*
BACK3   ASSIGN      6,V$LINE
        ADVANCE     *5
        RELEASE     *6
        INCREMENT   1,1
        LOOP        3,BACK3
*
        TABULATE    CTIME
SAI     TERMINATE   0
*
*   Relogio - tempo simulado
*

```


ANEXO 4

PROGRAMAS EM GPSS DO MODELO DE SIMULAÇÃO DO "WORMHOLE"

```
*
*
*   Nome      : WORMVETO.GPS
*   Estrategia : "wormhole"
*   Topologia  : vetor (10 processadores)
*   Autor     : Joao Paulo F. W. Kitajima
*
*
*   Alocação de espaço para transações
*
*   REALO     XAC,3000
*
*   Rotulo da função exponencial
EXPON EQU 1
*
*   Rotulo da tabela de tempos de comunicação
CTIME EQU 2
*
*   Tempo médio entre geração de mensagens
INTER EQU 3
*
*   Tempo de transmissão de um "flit"
TFLIT EQU 4
*
*   Numero de "flits" da mensagem
TMSG EQU 5
*
*   Tempo de transmissão dos "flits" restantes
TEMP EQU 6
*
*   Numero de "hops"
HOPS EQU 7
*
*   Tamanho mínimo de mensagem
HMIN EQU 8
```

```

*
*   Funcao exponencial
EXPON  FUNCTION  RN$1,C24
       0,0/.100,.104/.200,.222/.300,.355/.400,.509
       .500,.690/.600,.915/.700,1.200/.750,1.380
       .800,1.600/.840,1.830/.880,2.120/.900,2.300
       .920,2.520/.940,2.810/.950,2.990/.960,3.200
       .970,3.500/.980,3.900/.990,4.600/.995,5.300
       .998,6.200/.999,7/.9998,8
*
*   Escolhe um tamanho variavel de mensagem
TMSG  FVARIABLE  FN$EXPON*2001
*
*   Calcula tempo de comunicacao de "flits" restantes
*   ate a primeira liberacao de canal
TEMP  VARIABLE   (P$5 - (X$HOPS + 1))*X$TFLIT
*
*   Calcula tamanho minimo de mensagem
HMIN  VARIABLE   X$HOPS + 1
*
*   Define uma tabela de tempos de comunicacao
CTIME  TABLE M$1,0,20000,5
*
*   Inicializa tempo entre geracoes de mensagens
INITIAL  X$INTER,100000
*
*   Inicializa tempo de transmissao de um "flit"
INITIAL  X$TFLIT,5
*
*   Inicializa numero de "hops"
INITIAL  X$HOPS,2
*
SIMULATE
*
GENERATE  X$INTER, FN$EXPON, , , , 5, F
ASSIGN   1, 1
TRANSFER , HOP

```

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 2
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 3
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 4
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 5
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 6
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 7
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 8
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 9
 TRANSFER , HOP

*
 GENERATE X\$INTER, FN\$EXPON, , , , 5, F
 ASSIGN 1, 10
 TRANSFER , HOP

```

*
*
HOP  ASSIGN    3,X$HOPS
      ASSIGN    2,P$1
      ASSIGN    4,P$3
      ASSIGN    5,V$TMSG

*
      TEST_LE   P$5,X$HOPS,BACK
      ASSIGN    5,V$HMIN

*
*   Transmite primeiro "flit"
*
BACK QUEUE     *1
      SEIZE     *1
      DEPART    *1
      INCREMENT 1,1
      ADVANCE   X$TFLIT
      LOOP      3,BACK

*
      ASSIGN    1,P$2
      ASSIGN    3,P$4

*
      ASSIGN    5,V$TEMP
      ADVANCE   *5

*
*   Transmite "flits" restantes
*
BACK2 ADVANCE   X$TFLIT
      RELEASE   *1
      INCREMENT 1,1
      LOOP      3,BACK2

*
      TABULATE  CTIME
      TERMINATE 0

*
*   Relogio - tempo simulado
*

```



```

*
*   Nome      : WORMANEL.GPS
*   Estrategia : "wormhole"
*   Topologia  : anel (10 processadores)
*   Autor     : Joao Paulo F. W. Kitajima
*
*
*   Alocao de espaco para transacoes
*
*   REALD    XAC,3000
*
*   Rotulo da funcao exponencial
EXPON EQU    1
*
*   Rotulo da tabela de tempos de comunicacao
CTIME EQU    2
*
*   Tempo medio entre geracao de mensagens
INTER EQU    3
*
*   Tempo de transmissao de um "flit"
TFLIT EQU    4
*
*   Numero de "flits" da mensagem
TMSG EQU    5
*
*   Tempo de transmissao dos "flits" restantes
TEMP EQU    6
*
*   Linha corrente de transmissao
LINE EQU    7
*
*   Numero de "hops"
HOPS EQU    8
*
*   Tamanho minimo de mensagem
HMIN EQU    8

```

```

*
*   Funcao exponencial
EXPON FUNCTION RN$1,C24
      0,0/.100,.104/.200,.222/.300,.355/.400,.509
      .500,.690/.600,.915/.700,1.200/.750,1.380
      .800,1.600/.840,1.830/.880,2.120/.900,2.300
      .920,2.520/.940,2.810/.950,2.990/.960,3.200
      .970,3.500/.980,3.900/.990,4.600/.995,5.300
      .998,6.200/.999,7/.9998,8
*
*   Escolhe um tamanho variavel de mensagem
TMSG FVARIABLE FN$EXPON*2001
*
*   Calcula tempo de comunicacao de "flits" restantes
*   ate a primeira liberacao de canal
TEMP VARIABLE (P$5 - (X$HOPS + 1))*X$TFLIT
*
*   Calcula tamanho minimo de mensagem
HMIN VARIABLE X$HOPS + 1
*
*   Ajusta numero da ligacao em um anel
LINE VARIABLE (P$1 @ 20) + 1
*
*   Define uma tabela de tempos de comunicacao
CTIME TABLE M$1,0,20000,5
*
*   Inicializa tempo entre geracao de mensagens
INITIAL X$INTER,100000
*
*   Inicializa tempo de transmissao de um "flit"
INITIAL X$TFLIT,5
*
*   Inicializa numero de "hops"
INITIAL X$HOPS,2
*
SIMULATE

```

```
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 1
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 2
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 3
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 4
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 5
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 6
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 7
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 8
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
ASSIGN 1, 9
TRANSFER , HOP
*
GENERATE X$INTER, FN$EXPON, , , , 6, F
```

```

        ASSIGN      1,10
        TRANSFER    ,HOP
*
*
*
HOP  ASSIGN      3,X$HOPS
      ASSIGN      2,P$1
      ASSIGN      4,P$3
      ASSIGN      5,V$TMSG
*
      TEST_LE     P$5,X$HOPS,BACK
      ASSIGN      5,V$HMIN
*
*      Transmite primeiro "flit"
*
*
BACK  ASSIGN      6,V$LINE
      QUEUE       *6
      SEIZE       *6
      DEPART      *6
      INCREMENT   1,1
      ADVANCE     X$TFLIT
      LOOP        3,BACK
*
      ASSIGN      1,P$2
      ASSIGN      3,P$4
*
*      Transmite "flits" restantes ate liberacao
*      do primeiro canal
*
*
      ASSIGN      5,V$TEMP
      ADVANCE     *5
*
*      Transmite ultimo "flit"
*
*
BACK2  ASSIGN     6,V$LINE
        ADVANCE   X$TFLIT
        RELEASE   *6

```

```

INCREMENT 1,1
LOOP      3,BACK2

```

*

```

TABULATE  CTIME
TERMINATE 0

```

*

* Relogio - tempo simulado

*

```

GENERATE 10000000

```

*

```

TB$2 - tempo de comunicacao - valor da tabela CTIME

```

*

```

FR$5 - utilizacao do canal 5

```

*

```

AC$1 - valor do relógio do simulador

```

```

PRINT FR$5,TB$2,AC$1

```

```

TERMINATE 1

```

*

*

* Período transiente

*

```

START 1

```

```

RESET

```

*

*

* Equilibrio

*

```

START 1

```

```

RESET

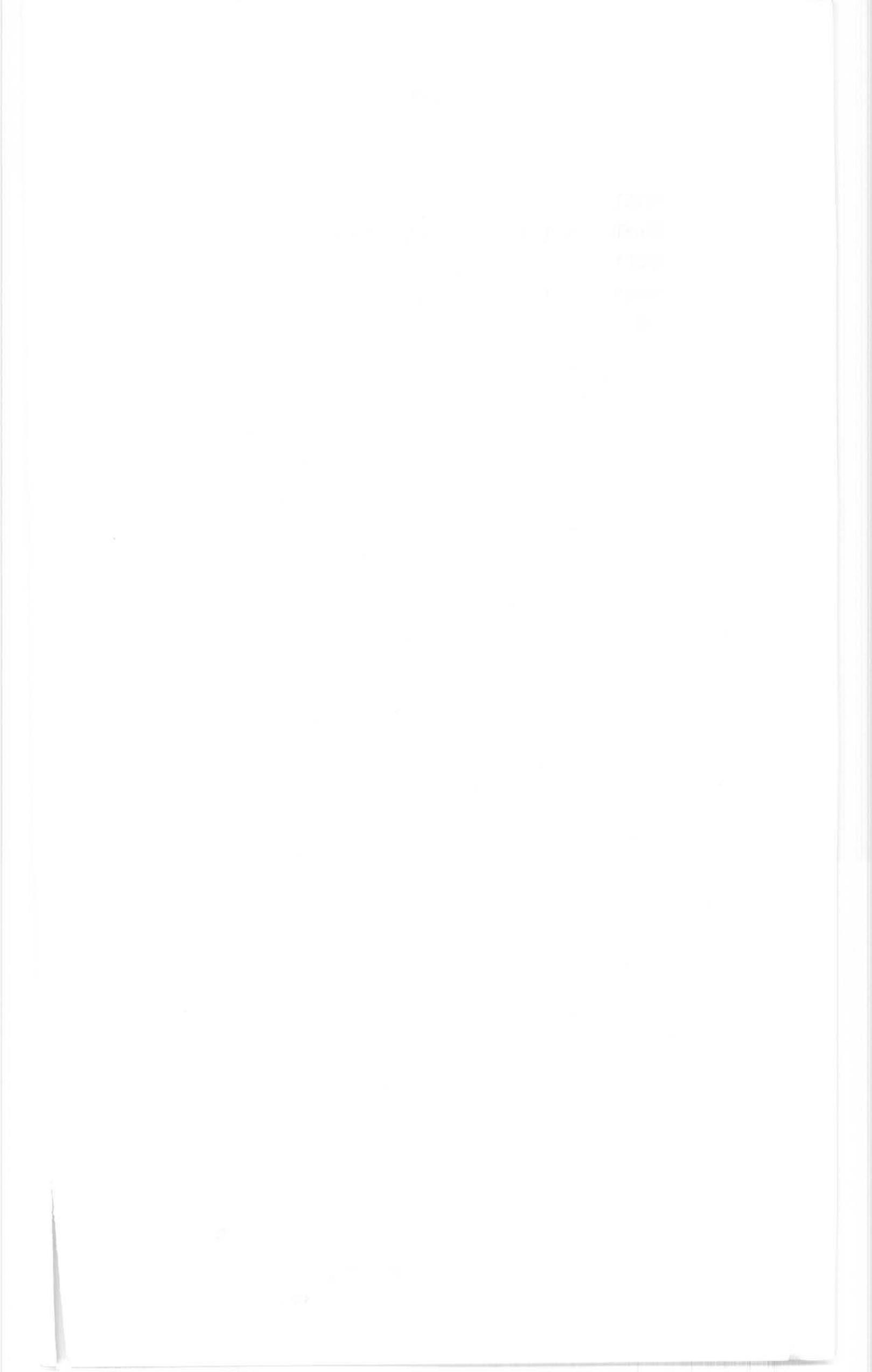
```

```

START 1

```

RESET
START 1
RESET
START 1
END



ANEXO 5

RESULTADOS NUMÉRICOS DAS SIMULAÇÕES EFETUADAS

Neste anexo, são apresentados os resultados numéricos das simulações efetuadas. A partir destes resultados, foram construídos os gráficos da validação do capítulo 7 (figura 7.4).

[Faint, illegible text, likely bleed-through from the reverse side of the page]

Intervalo de Utilização	Tempo de Comunicação	Tempo de Transmissão	Tempo de Comunicação	Tempo de Transmissão
0.00-0.10	100	100	100	100
0.10-0.20	150	150	150	150
0.20-0.30	200	200	200	200
0.30-0.40	250	250	250	250
0.40-0.50	300	300	300	300
0.50-0.60	350	350	350	350
0.60-0.70	400	400	400	400
0.70-0.80	450	450	450	450
0.80-0.90	500	500	500	500
0.90-1.00	550	550	550	550

Estratégia : rendez-vous
 Número de processadores : 10
 Topologia : vetor
 Tamanho da solicitação : 5 bits
 Tamanho do aceite (ACK) : 1 bit
 Tamanho da mensagem : 1000 bits
 Número de "hops" : 2

Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)

Tempo para alcance do
 equilíbrio : 10.000.000 unidades de tempo

Tempo para coleta de
 amostras : 10.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
50000	0.06+-0.00	2276.40+- 153.09	2208.86
25000	0.12+-0.00	2380.10+- 125.19	2451.37
20000	0.16+-0.01	2540.70+- 80.16	2644.95
10000	0.39+-0.01	4411.50+- 323.97	4844.31
9000	0.47+-0.01	6122.60+- 803.29	6815.15

Estratégia : rendez-vous

Número de processadores : 10

Topologia : vetor

Tamanho da solicitação : 5 bits

Tamanho do aceite (ACK) : 1 bit

Tamanho da mensagem : 5000 bits

Número de "hops" : 2

Número de amostras : 11 (a primeira amostra é desprezada - equilíbrio)

Tempo para alcance do equilíbrio : 20.000.000 unidades de tempo

Tempo para coleta de amostras : 20.000.000 unidades de tempo

Intervalo entre geração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
200000	0.07+-0.01	10753.70+- 980.59	11184.44
100000	0.16+-0.01	12776.40+- 392.94	13171.26
70000	0.23+-0.01	14152.90+- 520.10	15282.79
60000	0.30+-0.02	17463.40+- 1671.42	18200.55
50000	0.38+-0.02	22263.80+- 2343.33	23279.95

Estratégia : rendez-vous
 Número de processadores : 10
 Topologia : vetor
 Tamanho da solicitação : 5 bits
 Tamanho do aceite (ACK) : 1 bit
 Tamanho da mensagem : 10000 bits
 Número de "hops" : 2

Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)

Tempo para alcance do
 equilíbrio : 30.000.000 unidades de tempo

Tempo para coleta de
 amostras : 30.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
300000	0.10+-0.01	24154.20+- 1982.83	23541.29
200000	0.15+-0.01	24834.60+- 1248.87	25819.55
150000	0.22+-0.01	27670.80+- 1348.25	29866.00
130000	0.26+-0.01	29088.90+- 1840.28	32803.70
120000	0.29+-0.01	33078.10+- 2394.42	35416.44

Estratégia : rendez-vous

Número de processadores : 15

Topologia : vetor

Tamanho da solicitação : 5 bits

Tamanho do aceite (ACK) : 1 bit

Tamanho da mensagem : 1000 bits

Número de "hops" : 3

Número de amostras : 11 (a primeira amostra é desprezada - equilíbrio)

Tempo para alcance do equilíbrio : 10.000.000 unidades de tempo

Tempo para coleta de amostras : 10.000.000 unidades de tempo

Intervalo entre ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
100000	0.06+-0.01	3359.20+- 226.65	3426.34
50000	0.12+-0.01	3520.20+- 212.33	3967.58
40000	0.17+-0.01	4094.30+- 261.63	4568.95
30000	0.25+-0.01	4779.20+- 265.51	6031.50
20000	0.52+-0.02	13617.30+- 1456.92	-

Estratégia : rendez-vous
 Número de processadores : 15
 Topologia : vetor
 Tamanho da solicitação : 5 bits
 Tamanho do aceite (ACK) : 1 bit
 Tamanho da mensagem : 5000 bits
 Número de "hops" : 3

Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)

Tempo para alcance do
 equilíbrio : 20.000.000 unidades de tempo

Tempo para coleta de
 amostras : 20.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
200000	0.16+-0.01	19016.60+- 1525.80	22081.59
150000	0.24+-0.02	23145.90+- 2030.92	28876.38
130000	0.30+-0.01	30727.50+- 5243.11	37539.75
120000	0.35+-0.02	35388.20+- 5409.88	50053.50
100000	0.53+-0.04	72524.30+-15829.84	-

Estratégia : rendez-vous
 Número de processadores : 15
 Topologia : vetor
 Tamanho da solicitação : 5 bits
 Tamanho do aceite (ACK) : 1 bit
 Tamanho da mensagem : 10000 bits
 Número de "hops" : 3
 Número de amostras : 11 (a primeira amostra é desprezada - equilíbrio)
 Tempo para alcance do equilíbrio : 30.000.000 unidades de tempo
 Tempo para coleta de amostras : 30.000.000 unidades de tempo

Intervalo entre geração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
700000	0.08+-0.01	32126.20+- 4479.82	35732.43
500000	0.13+-0.01	36480.10+- 3482.76	40561.34
350000	0.19+-0.01	43186.30+- 5117.38	48412.21
300000	0.25+-0.02	48944.30+- 8043.06	60031.50
250000	0.34+-0.03	67433.40+-19132.06	93800.06

Estratégia : wormhole
 Número de processadores : 10
 Topologia : vetor
 Tamanho do "flit" : 5 bits
 Tamanho da mensagem : 201 "flits"
 Número de "hops" : 2

Número de amostras : 11 (a primeira amostra é desprezada - equilíbrio)

Tempo para alcance do equilíbrio : 10.000.000 unidades de tempo

Tempo para coleta de amostras : 10.000.000 unidades de tempo

Intervalo entre geração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
50000	0.03+-0.00	1041.80+- 68.71	1073.14
20000	0.10+-0.00	1184.30+- 34.37	1246.47
10000	0.21+-0.01	1444.10+- 56.62	1626.20
7000	0.33+-0.01	1962.70+- 61.45	2323.47
6000	0.42+-0.01	2698.40+- 239.70	3291.62

Estratégia : wormhole

Número de processadores : 10

Topologia : vetor

Tamanho do "flit" : 5 bits

Tamanho da mensagem : 1001 "flits"

Número de "hops" : 2

Número de amostras : 11 (a primeira amostra é
desprezada - equilíbrio)

Tempo para alcance do
equilíbrio : 20.000.000 unidades de tempo

Tempo para coleta de
amostras : 20.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
100000	0.10+-0.01	5801.60+- 365.04	6187.65
80000	0.13+-0.01	6242.40+- 309.81	6626.52
60000	0.18+-0.01	6715.30+- 364.93	7478.22
50000	0.21+-0.01	6752.50+- 284.85	8078.76
30000	0.40+-0.01	12101.70+- 1344.55	15020.00

Estratégia : wormhole
 Número de processadores : 10
 Topologia : vetor
 Tamanho do "flit" : 5 bits
 Tamanho da mensagem : 2001 "flits"
 Número de "hops" : 2

Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)

Tempo para alcance do
 equilíbrio : 30.000.000 unidades de tempo

Tempo para coleta de
 amostras : 30.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
200000	0.10+-0.01	11363.80+- 1025.76	12364.12
150000	0.13+-0.01	12224.40+- 715.82	13241.43
100000	0.21+-0.01	14283.90+- 1417.77	16144.45
75000	0.30+-0.01	18164.30+- 1040.62	20924.55
60000	0.41+-0.01	25488.50+- 2606.21	31319.35

Estratégia : wormhole
 Número de processadores : 15
 Topologia : vetor
 Tamanho do "flit" : 5 bits
 Tamanho da mensagem : 201 "flits"
 Número de "hops" : 3

Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)
 Tempo para alcance do
 equilíbrio : 10.000.000 unidades de tempo
 Tempo para coleta de
 amostras : 10.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
75000	0.03+-0.00	1026.70+-	74.86 1111.22
30000	0.10+-0.01	1179.00+-	78.52 1391.88
20000	0.15+-0.00	1303.60+-	41.01 1724.41
15000	0.22+-0.00	1597.30+-	52.18 2199.46
10000	0.39+-0.01	2662.80+-	108.19 6359.77

Estratégia : wormhole
 Número de processadores : 15
 Topologia : vetor
 Tamanho do "flit" : 5 bits
 Tamanho da mensagem : 1001 "flits"
 Número de "hops" : 3

Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)

Tempo para alcance do
 equilíbrio : 20.000.000 unidades de tempo

Tempo para coleta de
 amostras : 28 888 888 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
200000	0.07+-0.01	5450.00+- 414.09	6327.15
80000	0.20+-0.01	7338.80+- 507.66	10020.00
60000	0.29+-0.01	9343.20+- 975.95	15382.50
50000	0.40+-0.02	14203.90+- 2330.48	35045.00
45000	0.48+-0.02	19098.90+- 3135.37	185195.00

Estratégia : wormhole
 Número de processadores : 15
 Topologia : vetor
 Tamanho do "flit" : 5 bits
 Tamanho da mensagem : 2001 "flits"
 Número de "hops" : 3

 Número de amostras : 11 (a primeira amostra é
 desprezada - equilíbrio)
 Tempo para alcance do
 equilíbrio : 30.000.000 unidades de tempo
 Tempo para coleta de
 amostras : 30.000.000 unidades de tempo

Intervalo entre ge- ração de mensagens	Utilização	Tempo de Comunicação (simulação)	Tempo de Comunicação (analítico)
400000	0.07+-0.01	10712.90+- 798.08	12458.08
200000	0.16+-0.01	13083.80+- 690.06	17077.35
125000	0.28+-0.01	17725.80+- 1559.99	29115.45
100000	0.39+-0.01	25894.40+- 2903.12	63223.41
90000	0.49+-0.02	36806.40+- 3799.49	743382.50

1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900

1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930

BIBLIOGRAFIA

- [BER 89] BERTSEKAS, Dimitri P. & TSITSIKLIS, John N. Parallel and distributed computation: numerical methods. Englewood Cliffs, Prentice-Hall, 1989.
- [BOK 81] BOKHARI, Shahid H. On the mapping problem. IEEE Transactions on Computers, New York, 30(3):207-14, Mar. 1981.
- [CHA 84] CHANG, Jo-mei & MAXEMCHUK, N. F. Reliable broadcast protocols. ACM Transactions on Computer Systems, New York, 2(3):251-73, Aug. 1984.
- [DAL 87] DALLY, William J. & SEITZ, Charles L. Deadlock-free message routing in multiprocessor interconnection networks. IEEE Transactions on Computers, New York, 36(5):547-53, May 1987.
- [DON 79] O'DONOVAN, Thomas M. GPSS : simulation made simple. Chichester, Wiley, 1979.
- [EAG 86] EAGER, Derek L.; LAZOWSKA, Edward D. & ZAHORJAN, John. A comparison of receiver-initiated and sender-initiated adaptive load sharing. Performance Evaluation, Amsterdam, 6(1):53-67, Mar. 1986.
- [FOX 87] FOX, Geoffrey & MESSINA, Paul C. Advanced computer architectures. Scientific American, New York, 257(4):44-52, Oct. 1987.
- [GPS 83] GPSSR/PC introduction and reference manual. London, 1983. (version 2.0).

- [HWA 84] HWANG, Kai & BRIGGS, Fayé A. Computer architecture and parallel processing. New York, McGraw-Hill, 1984.
- [JAC 80] JACOBY, Samuel L. S. & KOWALIK, Janusz S. Mathematical modeling with computers. Englewood Cliffs, Prentice-Hall, 1980.
- [JES 88] JESSHOPE, Chris. Transputers and switches as objects in Occam. Parallel Computing, Amsterdam, 8(1-3):19-30, Oct. 1988.
- [KER 79] KERMANI, Paviz & KLEINROCK, Leonard. Virtual cut-through: a new computer communication switching technique. Computer Networks, Amsterdam, 3(4):267-86, Sept. 1979.
- [KLE 75] KLEINROCK, Leonard. Queueing systems. New York, Wiley, 1975. v.1.
- [KLE 76] KLEINROCK, Leonard. Queueing systems. New York, Wiley, 1976. v.2.
- [KOB 78] KOBAYASHI, Hisashi. Modeling and analysis: an introduction to system performance evaluation methodology. Reading, Addison-Wesley, 1978.
- [MAR 86] MARSAN, M. Ajmone; BALBO, G. & CONTE, G. Performance models of multiprocessor systems. Cambridge, The MIT Press, 1986.
- [PLA 89] PLATEAU, Brigitte. Tempos de comunicação em multiprocessadores: notas do estágio. Grenoble, 1989.

- [SCH 67] SCHERR, A. L. An analysis of time-shared computer systems. Cambridge, The MIT Press, 1967.
- Apud FERRARI, Domenico. Computer systems performance evaluation. Englewood Cliffs, Prentice-Hall, 1978.
- [SEI 89] SEITZ, Charles L. An assessment of second-generation multicomputers. In: HYPERCUBE AND DISTRIBUTED COMPUTERS, Rennes, Oct. 4-6, 1989. Proceedings. Amsterdam, North-Holland, 1989. p. 1.
- [TRY 89] TRYSTRAM, D. & VINCENT, F. Programmation avancée du TRANSPUTER: architecture et mécanismes. La Lettre du Transputer: 9-18, Avr. 1989.
- [WIL 86] WILLIAMS, Andrew T. Lotus 1-2-3 de A a Z: com ligações para o symphony. Rio de Janeiro, Livros Técnicos e Científicos, 1986.