
Stanley Robson de Medeiros Oliveira

**ALLOS - Uma ferramenta para
solucionar modelos de redes de filas
usando cadeias de Markov**

Campina Grande - PB

Junho de 1995

Stanley Robson de Medeiros Oliveira

**ALLOS - Uma ferramenta para
solucionar modelos de redes de filas
usando cadeias de Markov**

Dissertação apresentada ao Curso de Mestrado em Informática da Universidade Federal da Paraíba, como exigência parcial para a obtenção do Grau de Mestre.

Área de concentração: Ciência da Computação

Orientadores: Prof. Marcos Antônio Gonçalves Brasileiro
e Prof^a Maria Izabel Cavalcanti Cabral
Universidade Federal da Paraíba - UFPB

Campina Grande - PB

1995



048a Oliveira, Stanley Robson de Medeiros
ALLOS : uma ferramenta para solucionar modelos de redes de filas usando cadeias de Markov / Stanley Robson de Medeiros Oliveira. - Campina Grande, 1995.
56 f. : il.

Dissertacao (Mestrado em Informatica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Modelo de Rede - 2. Avaliacao de Desempenho 3. Geracao Automatica de Estados 4. Cadeias de Markov 5. Redes de Filas 6. Dissertacao I. Brasileiro, Marcos Antonio Goncalves, Dr. II. Cabral, Maria Izabel Cavalcanti, Dra. III. Universidade Federal da Paraiba - Campina Grande (PB) IV. Título

CDU 004.625.3:519.217.2(043)

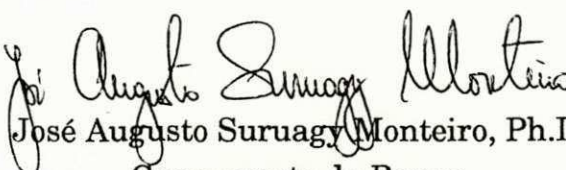
**ALLOS - Uma ferramenta para
solucionar modelos de redes de filas
usando cadeias de Markov**


Stanley Robson de Medeiros Oliveira

Dissertação aprovada em 21 de junho de 1995


Marcos Antônio Gonçalves Brasileiro, Dr.
Orientador


Maria Izabel Cavalcanti Cabral, Dr.^a.
Orientadora


José Augusto Suruagy Monteiro, Ph.D
Componente da Banca


José Antão Beltrão Moura, Ph.D
Componente da Banca

Campina Grande, Pb

A meus pais, Severino Luís e Saete Medeiros
A meus irmãos Stael, Suleyman (in memorian), Sundar, Smith e Shary

Agradecimentos

Agradeço ao Senhor Deus que, dia-a-dia, leva o meu fardo, renova as minhas forças e me conduz em triunfo, por meio de Cristo, permitindo que os meus sonhos se tornem história. Ele é a pessoa mais interessada na minha vida

À minha família abençoada, em particular a meus pais, Severino Luís e Salete Medeiros; a meus irmãos Stael, Suleyman (in memorian), Sundar, Smith e Shary, pela confiança transmitida, pelo incentivo incansável, pela compreensão e assitência contínua durante a realização deste trabalho

Ao meu orientador, professor Marcos Brasileiro, que me transmitiu segurança, dando-me toda atenção e me incentivando, sobretudo se dispondo a servir

À minha orientadora, professora Maria Izabel, pela confiança e segurança transmitidas, pelo grande incentivo, por me introduzir na comunidade científica, por toda a atenção dispensada em meu favor e, sobretudo, pela alegria que tem em poder servir

Ao incansável amigo e professor Edilson Ferneda, pela ajuda, sugestões valiosíssimas e todo incentivo, pela diagramação deste texto e, especialmente, pela grande contribuição no capítulo 3 (geração automática de estados)

A Flávia, por tanta compreensão, carinho e incentivo em toda essa trajetória

Aos professores, membros da banca examinadora, Izabel, Marcos Brasileiro, Antão e Suruagy, por fazerem parte da banca e, especialmente, pela segurança transmitida e valiosas sugestões que contribuíram, sobremaneira, para o enriquecimento da redação final deste trabalho

Às colegas Elisabete e Nívia, pela revisão cuidadosa de todas as referências bibliográficas deste trabalho

A Nísia Leão, por sua dedicação na revisão deste texto, tornando-o legível e de fácil compreensão

Aos funcionários da Coordenação de Pós-Graduação em Informática, Aninha, Vera e Marcelo, pela atenção e disponibilidade para servir

Aos professores do Departamento de Sistemas e Computação, em especial Izabel, Edilson, Hamurabi, Antão, Hélio, Giuseppe, Peter e Hattori pelo estímulo, segurança e maneira como cada um deles me transmitiu o conhecimento

Sinceros agradecimentos às instituições:

- Empresa Brasileira de Pesquisa Agropecuária - EMBRAPA, por suas ações prioritárias que promoveram a minha capacitação e o meu aperfeiçoamento para a realização de seus próprios objetivos, pela valorização e incentivo dispensados em meu favor
- À Coordenação de Pós-Graduação em Informática, pela atenção e subsídio na participação da Conferência Internacional (IASTED AMSO'95) realizada em Cancún, México
- À CAPES, pelo apoio financeiro para conduzir este trabalho de pesquisa nos dois primeiros anos
- À Universidade Federal da Paraíba, por meio do Centro de Ciências e Tecnologia pelo subsídio na participação da Conferência Internacional (IASTED AMSO'95) realizada em Cancún, México

Finalmente, os meus sinceros agradecimentos a todos aqueles que envidaram esforços no sentido de colaborar com a realização deste trabalho.

Sumário

Resumo

1	Introdução	1
1.1	Objetivo	4
1.2	Motivação e contribuição científica	4
1.3	Organização da dissertação	5
2	O processo de modelagem usando <i>ALLOS</i>	7
2.1	Estrutura do <i>SAVAD</i>	7
2.2	Elementos de modelagem	9
2.3	Interface do <i>SAVAD</i>	13
2.4	Medidas de desempenho	15
2.5	Exemplo de especificação de um modelo	18
3	<i>ALLOS</i> - Concepção e desenvolvimento	21
3.1	Geração automática de estados	21
3.2	Procedimentos de geração automática de estados	23
3.3	Solução da matriz de taxas de transição de estados	25
3.4	Estrutura de <i>ALLOS</i>	28
3.5	Desenvolvimento de <i>ALLOS</i>	30
4	Validação	32
4.1	Validação através de Técnicas Analíticas	32
4.1.1	Sistema M/M/1/5	32
4.1.2	Sistema M/M/3/10	34
4.1.3	Modelo de uma Rede Aberta sem Realimentação	35
4.1.4	Modelo de uma Rede Fechada	36
4.1.5	Modelo de uma Rede Aberta com Ponto de Multiplicação	38

4.1.6	Modelo de uma Rede Aberta com Ponto de Fusão	40
4.1.7	Modelo do Protocolo de Sessão com Diálogo Semi-Duplex	42
4.1.8	Modelo do Protocolo de Sessão com Quarentena de Dados e Modo de Diálogo Duplex	45
4.2	Validação através de Simulação Digital	47
4.2.1	Modelo de uma Rede Local Anel com Passagem de Ficha	47
5	Conclusões e sugestões	50
	Abstract	53
	Referências bibliográficas	54

Lista de figuras

FIGURA 2.1 - Módulos do SAVAD	8
FIGURA 2.2 - Elementos de modelagem do tipo <i>estação de serviço</i>	10
FIGURA 2.3 - Elemento de modelagem do tipo <i>fonte</i>	10
FIGURA 2.4 - Elemento de modelagem do tipo <i>sorvedouro</i>	11
FIGURA 2.5 - Elementos de modelagem do tipo <i>ponto de controle</i>	11
FIGURA 2.6 - Interface básica de SAVAD	15
FIGURA 2.7 - Modelo de uma rede cíclica	18
FIGURA 2.8 - Especificação do elemento de modelagem UCP	19
FIGURA 2.9 - Especificação do elemento de modelagem E/S	19
FIGURA 2.10 - Especificação do elemento de modelagem Rota	19
FIGURA 2.11 - Medidas de desempenho para o modelo de uma rede cíclica	20
FIGURA 2.12 - Vetor de probabilidades para o modelo de uma rede cíclica	20
FIGURA 2.13 - Diagrama de transições entre estados para o modelo de uma rede cíclica	20
FIGURA 3.1 - Possíveis transições entre os estados dos modelos	23
FIGURA 3.2 - Diagrama de transições entre estados para o modelo de uma rede cíclica	28
FIGURA 3.3 - Estrutura de ALLOS no ambiente SAVAD	28
FIGURA 3.4 - Arquivo com a base de fatos para o modelo de uma rede cíclica	29
FIGURA 3.5 - Arquivo <i>espest.mrk</i> para o modelo da rede cíclica	30
FIGURA 3.6 - Arquivo <i>transic.mrk</i> para o modelo da rede cíclica	30
FIGURA 4.1 - Modelo para o Sistema M/M/1/5	32
FIGURA 4.2 - Vetor de probabilidades para o modelo do sistema M/M/1/5	33
FIGURA 4.3 - Modelo para o Sistema M/M/3/10	34
FIGURA 4.4 - Vetor de probabilidades para o modelo do sistema M/M/3/10	34
FIGURA 4.5 - Modelo de uma rede aberta sem realimentação	35
FIGURA 4.6 - Vetor de probabilidades para o modelo de uma rede aberta sem realimentação	36
FIGURA 4.7 - Modelo de uma rede fechada	37

FIGURA 4.8 - Vetor de probabilidades para o modelo de uma rede fechada	37
FIGURA 4.9 - Rede aberta com ponto de multiplicação	38
FIGURA 4.10 - Vetor de probabilidades para o modelo de uma rede aberta com ponto de multiplicação	39
FIGURA 4.11 - Modelo de uma rede aberta com ponto de fusão	40
FIGURA 4.12 - Vetor de probabilidades para o modelo de uma rede aberta com ponto de fusão	41
FIGURA 4.13 - Modelo do protocolo de sessão com diálogo semi-duplex	43
FIGURA 4.14 - Vetor de probabilidades para o modelo do protocolo de sessão com diálogo semi-duplex	44
FIGURA 4.15 - Modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex	45
FIGURA 4.16 - Modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex	46
FIGURA 4.17 - Modelo de uma rede local anel com passagem de ficha	47
FIGURA 4.18 - Vetor de probabilidades para o modelo de uma rede local anel com passagem de ficha	49

Lista de tabelas

TABELA 4.1 -	Medidas de desempenho para o modelo do sistema M/M/1/5	33
TABELA 4.2 -	Medidas de desempenho para o modelo do sistema M/M/3/10	35
TABELA 4.3 -	Medidas de desempenho para o modelo de uma rede aberta sem realimentação	36
TABELA 4.4 -	Medidas de desempenho para o modelo de uma rede fechada	37
TABELA 4.5 -	Medidas de desempenho para o modelo de uma rede aberta com ponto de multiplicação	40
TABELA 4.6 -	Medidas de desempenho para o modelo de uma rede aberta com ponto de fusão	42
TABELA 4.7 -	Medidas de desempenho para o modelo do protocolo de sessão com diálogo semi-duplex	44
TABELA 4.8 -	Medidas de desempenho para o modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex	46
TABELA 4.9 -	Medidas de desempenho para o modelo de uma rede local anel com passagem de ficha	49

Resumo

A relevância de modelos markovianos na avaliação de desempenho de sistemas e a necessidade de ferramentas eficientes para a especificação e solução de modelos tornam-se o alvo das atenções da nova geração de sistemas concorrentes. Estas ferramentas devem ser sofisticadas ao ponto de tornar transparentes aos seus usuários, a definição matemática do sistema modelado e os detalhes inerentes às técnicas de solução do modelo.

É neste sentido que esta dissertação apresenta *ALLOS*, uma ferramenta amigável para solucionar modelos de redes de filas usando cadeias de Markov. O processo de geração do espaço de estados para um modelo específico é feito automaticamente, a partir de um estado inicial e das regras que descrevem o comportamento do sistema.

ALLOS faz parte de um ambiente inteligente e integrado para modelar e solucionar sistemas de redes de filas denominado *SAVAD*, um projeto multidisciplinar que envolve os Grupos de Redes de Computadores e de Inteligência Artificial da Universidade Federal da Paraíba - CCT/UFPB.

O uso de *ALLOS* direciona-se, principalmente, aos sistemas que exibem contenção de recursos, a saber, sistemas de computação e redes de computadores, além de outros que podem ser modelados utilizando o paradigma de redes de filas.

1 Introdução

A complexidade da nova geração de sistemas altamente concorrentes que estão sendo atualmente projetados, impõe o desenvolvimento de sofisticadas ferramentas de modelagem para a análise de desempenho de tais sistemas [DINIZ, 1991]. A modelagem e a análise de sistemas de computação têm recebido atenção cada vez maior, por parte de pesquisadores e projetistas que desejam entender e prever o comportamento desses sistemas [SILVA, 1992b].

Um dos procedimentos adotados na análise de desempenho de um sistema é a obtenção de uma abstração que englobe o seu comportamento. Esta abstração é chamada modelo do sistema. Um modelo é uma visão simplificada do sistema em estudo, porém projetado com o propósito de capturar o máximo do seu comportamento real. Em muitos casos, modelos possuem certas vantagens sobre medidas feitas em um sistema real. Algumas vezes, um modelo é a única alternativa concreta quando o sistema ainda não foi desenvolvido [SILVA, 1992b].

Existem sistemas que apresentam contenção de recursos como, por exemplo, sistemas de computação, redes de computadores, controle de processos, sistemas de manufatura e sistemas de tráfego, entre outros, os quais podem ser modelados através do paradigma de redes de filas [KLEINROCK, 1975; SAUER, 1981]. Uma rede de filas é um sistema em que se considera a existência de múltiplas filas operando assíncrona e concorrentemente, interconectadas conforme uma topologia específica. Cada sistema de filas é constituído por um processo estocástico de chegada de clientes, por um processo estocástico de serviço para cada cliente, pelo número de servidores do sistema, pelas classes de clientes que determinam os diferentes tipos de clientes no sistema e por uma ordem pela qual esses clientes são atendidos [MOURA, 1986].

A solução de modelos de redes de filas ainda é uma tarefa complexa, sendo restrita apenas àqueles que possuem conhecimento das técnicas disponíveis na literatura especializada. Para solucionar tais modelos, pode-se utilizar técnicas analíticas, que são mais econômicas e eficientes; contudo, muitas vezes exigem a simplificação do

modelo em virtude de sua complexidade. Outra alternativa são as técnicas aproximadas, onde o custo computacional pode ser um fator limitante.

Uma das alternativas na modelagem de sistemas de redes de filas é o uso de processos markovianos. Um processo é markoviano se, dado o estado presente, o comportamento futuro do processo é estatisticamente independente do seu passado, isto é, a história do sistema é sumarizada através do seu estado presente [SILVA, 1992b; KOBAYASHI, 1978]. Esta classe de processos é a mais amplamente usada na análise e na modelagem de sistemas de computação/comunicação, e é suficientemente adequada na maioria dos casos [DINIZ, 1990].

Tradicionalmente, utilizavam-se, na solução de sistemas de redes de filas, as técnicas analíticas, quando possível, ou se empregavam linguagens de propósito geral, tais como GPSS [SHRIBER, 1974] e SIMULA [BIRTWISTLE, 1973]. Percebe-se, hoje, uma tendência no uso de ambientes integrados que fornecem soluções analíticas e aproximadas e, além disso, apresentam capacidade de desenvolvimento de modelos complexos, permitindo aos usuários a construção, a modificação e a representação de modelos do mundo real. Exemplos de alguns desses ambientes é SAVAD [CABRAL, 1994; SOUTO, 1993], RESQ [SAUER, 1985], BONes e OPNET [MCCOMAS, 1994].

Para a construção de tais ambientes propõem-se técnicas estruturadas, facilidade de interação com o ambiente computacional, componentes de software reutilizáveis e facilidade de extensão de bibliotecas de módulos de software [BOOCH, 1991; STROUSTRUP, 1991].

SAVAD (Sistema de AVAliação de Desempenho de Modelos de Redes de Filas) é um projeto multidisciplinar que envolve os Grupos de Redes de Computadores e de Inteligência Artificial do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba - CCT/UFPB. Este sistema foi desenvolvido usando as Linguagens de Programação C++ [STROUSTRUP, 1991] e Prolog [CLOCKSIN, 1981] em sistemas de microcomputadores compatíveis com IBM-PC.

SAVAD é um ambiente inteligente e integrado para modelagem e avaliação de desempenho de redes de filas. Ele é, basicamente, constituído por três módulos: *Interface*, *Núcleo do Sistema Especialista* e *Solução*. A abordagem orientada a objetos é utilizada no projeto e no desenvolvimento deste sistema, uma vez que esta abordagem apresenta vantagens significativas, tais como modularidade, extensibilidade e reutilização de "software".

A partir de um modelo de redes de filas proposto pelo usuário, SAVAD procura uma solução para o modelo e fornece as medidas de desempenho mais relevantes. Para tanto, o sistema deve escolher adequadamente as técnicas utilizadas, sejam elas exatas ou aproximadas. Entre as técnicas exatas, ressalta-se, nesta dissertação, a

ferramenta *ALLOS* que gera automaticamente os estados dos modelos markovianos, solucionando-os, a fim de obter as medidas de desempenho mais relevantes. O processo de geração do espaço de estados de um modelo a ser solucionado é feito automaticamente a partir de um estado inicial e das regras que descrevem o comportamento deste modelo.

Diversas ferramentas têm sido desenvolvidas para analisar o desempenho ou a confiabilidade de sistemas de computação [DINIZ, 1990; MAKAN, 1982; CHIOLA, 1985; GOYAL, 1986; JOHNSON JR., 1988; CIARDO, 1989; PAGE JUNIOR, 1989]. A maioria dessas ferramentas usa modelos markovianos como modelo matemático que descreve o comportamento do sistema sendo analisado. Em [BERSON, 1990] é proposto um novo paradigma para especificação de modelos (modelos orientados a objetos) e implementado na ferramenta *TANGRAM* [PAGE JUNIOR, 1989] desenvolvida na UCLA (Universidade da Califórnia, Los Angeles) e na ferramenta co-irmã desenvolvida na UFRJ [DINIZ, 1990]. Nestas ferramentas, o sistema a ser modelado é visto como uma coleção de objetos que interagem pela troca de mensagens. Cada objeto é uma entidade com um estado interno que evolui com o tempo. O estado de um objeto pode mudar em consequência de um evento gerado pelo próprio objeto ou de uma mensagem recebida de um outro objeto. O estado de um objeto determina os tipos de eventos que podem ocorrer e as taxas de ocorrência desses eventos. Portanto, a especificação de um objeto inclui a definição dos eventos que ele pode gerar, as ações tomadas quando da ocorrência de um evento e a descrição de como o objeto reage ao recebimento de mensagens. Maiores detalhes podem ser obtidos em [DINIZ, 1990; DINIZ, 1991].

No desenvolvimento de uma ferramenta de modelagem sofisticada, a definição matemática do modelo deve ser transparente ao usuário. A ferramenta também deve permitir a descrição de modelos mais próxima possível da definição natural do sistema [DINIZ, 1990].

A opção pelo uso de cadeias de Markov se dá pelo fato de que modelos markovianos têm causado impacto nos trabalhos de pesquisa concernentes à área de avaliação de desempenho de sistemas. Além disso, processos markovianos são ferramentas bastante úteis para avaliação de desempenho de sistemas, visto que simplificam bastante o tratamento analítico e são menos susceptíveis a erros quando comparados com simulações [MOURA, 1986]. Outro fator relevante é que um modelo markoviano pode ser alterado (e.g., truncando o espaço de estados) de tal forma que este se torne mais tratável. No entanto, alguns modelos markovianos possuem um espaço de estados cuja cardinalidade torna proibitiva a sua solução. Desta forma, a explosão de estados se constitui o maior problema que inibe a aplicação mais abrangente desta técnica [SILVA, 1992b].

Com o decréscimo no custo de memória de computadores, o aumento da capacidade de processamento e o avanço das técnicas para solução de modelos, processos markovianos com um grande número de estados podem, agora, ser solucionados, reduzindo a forte restrição imposta pela modelagem markoviana, que é a explosão de estados [SILVA, 1988].

1.1 Objetivo

A nova geração de sistemas impõe o uso de ferramentas de modelagem sofisticadas para a avaliação de desempenho. O desenvolvimento de tais ferramentas encontra-se em evidência, ultimamente, uma vez que estas permitem, aos seus usuários, facilmente construir e modificar representações de sistemas reais, sem a necessidade de conhecimento da representação matemática e sem a exigência de se conhecer, profundamente, as técnicas de modelagem disponíveis na literatura.

Neste sentido, a presente dissertação apresenta o desenvolvimento de uma ferramenta amigável para solucionar modelos de redes de filas usando cadeias de Markov. Esta ferramenta gera automaticamente os estados de modelos markovianos, a partir de um estado inicial e das regras que descrevem o comportamento do sistema, soluciona estes modelos, para, em seguida, apresentar as medidas de desempenho aos usuários. Além disso, *ALLOS* apresenta o espaço de estados e o vetor $\vec{P}_i(n)$, que contém a probabilidade de encontrar a cadeia no estado i no tempo n . Em regime estacionário, ou seja, quando n tende para infinito, as probabilidades da cadeia se encontrar no estado $i = 0, 1, 2, \dots, N$ são fornecidas pelo limite de $\vec{P}(n)$ [MOURA, 1986].

1.2 Motivação e contribuição científica

Conforme discutido na introdução deste trabalho, várias ferramentas têm sido desenvolvidas para analisar o desempenho e a confiabilidade de sistemas. *ALLOS* centraliza seu domínio de aplicação na avaliação de desempenho de sistemas de redes de filas, supondo-se que estes sejam perfeitamente confiáveis.

Modelos markovianos têm sido bastante usados na avaliação de desempenho de sistemas, por parte de analistas que desejam entender e prever o comportamento de tais sistemas [SILVA, 1992b]. No entanto, o uso desta técnica analítica encontra-se, geralmente, disponível somente para os especialistas. Neste sentido, o desenvolvimento de uma ferramenta de modelagem sofisticada torna-se imprescindível para a modelagem e avaliação de desempenho de sistemas que exibem contenção de recursos, onde a ênfase se concentra nas facilidades que a ferramenta oferece aos seus usuários, não exigindo destes o conhecimento de um especialista.

A relevância do uso de modelos markovianos na avaliação de desempenho e a necessidade de ferramentas eficientes para a especificação e solução de modelos, acima mencionadas, tornaram-se razões bastante significativas para o desenvolvimento de *ALLOS*. Desta forma, a contribuição desta dissertação fundamenta-se nos seguintes aspectos:

- Uso da técnica de exploração dinâmica para guiar o processo de geração da matriz de transição de estados. A importância de um estado deve ser dada em termos da sua contribuição para o cálculo de medidas de desempenho [DINIZ, 1991; SILVA, 1992a; SILVA, 1992b]. Uma forma elegante e eficiente para a exploração dinâmica é a geração automática de estados quando, a partir da especificação do modelo de um sistema por parte do usuário, *ALLOS* gera a matriz de transição de estados a partir de um estado inicial e das regras que descrevem o comportamento do sistema;
- A ferramenta proposta é mais uma opção para solucionar sistemas de redes de filas e está anexada ao módulo solução do *SAVAD*;
- Através da ferramenta *ALLOS* um usuário pode, facilmente, especificar o modelo de um sistema e obter as medidas de interesse mais relevantes, de forma automática, sem que este necessite conhecer os detalhes da representação matemática.
- Outra facilidade de *ALLOS* é a possibilidade de se obter o espaço de estados e o vetor de probabilidades do modelo, sendo analisado.
- São explorados alguns recursos da linguagem *Prolog* para a implementação do módulo de geração automática de estados, conforme discutidos na seção 3.5.
- *ALLOS* está disponível para funcionamento em equipamentos de pequeno porte, como microcomputadores compatíveis com a linha IBM/PC, com possibilidade de uso do modelo de geração automática em ambiente Unix. Para o ambiente operacional MS-DOS a solução de um modelo de redes de filas limita-se a um espaço de estados na ordem de 150 estados.

1.3 Organização da dissertação

Esta dissertação está organizada em 5 capítulos, conforme apresentados a seguir:

O capítulo 2 focaliza o processo de modelagem usando *ALLOS*, onde destacam-se, inicialmente, a estrutura do *SAVAD* e sua interação com a ferramenta *ALLOS*. A flexibilidade dos elementos de modelagem é apresentada neste capítulo, possibilitando a modelagem de uma grande variedade de sistemas que apresentam contenção de recursos. Além disso, é feita uma abordagem sucinta da interface de *ALLOS*, que é a

mesma utilizada pelo *SAVAD*, com ênfase dada à interação entre o usuário e o sistema, além das facilidades de manipulação de arquivos e acesso direto ao sistema operacional. O capítulo apresenta as medidas de desempenho fornecidas por *ALLOS* e encerra com um exemplo da especificação de um modelo por parte do usuário.

A concepção e o desenvolvimento de *ALLOS* são discutidos no capítulo 3, onde o alvo das atenções é o processo de geração automática de estados, juntamente com os procedimentos para a obtenção do espaço de estados associado a um modelo e a geração da matriz de taxas de transição entre os estados. A solução da matriz de taxas de transição de estados é apresentada neste capítulo, com a finalidade de se obter o vetor de probabilidades de estados para o cálculo das medidas de desempenho. O capítulo também focaliza a estrutura de *ALLOS* (seus módulos e suas funcionalidades) e finaliza apresentando considerações gerais sobre o desenvolvimento da ferramenta.

No capítulo 4 é mostrado o processo de validação da ferramenta *ALLOS*, onde são apresentados vários exemplos que evidenciam as potencialidades desta ferramenta. Os resultados dos modelos solucionados são comparados com os resultados obtidos na literatura especializada (técnicas analíticas [KLEINROCK, 1975; REISER, 1979; BURKE, 1956] e técnicas aproximadas, baseadas em simulação digital [CONCEIÇÃO FILHO, 1993]).

Finalmente, no capítulo 5 são apresentadas as conclusões e sugestões de continuidade deste trabalho.

2 O processo de modelagem usando *ALLOS*

Este capítulo é dedicado aos usuários de *ALLOS*, que não necessitam ser especialistas em análise de desempenho para solucionar modelos de redes de filas usando cadeias de Markov, visto que toda representação matemática utilizada neste processo de modelagem lhes é transparente. Inicialmente, a estrutura do *SAVAD* é apresentada com a finalidade de mostrar a interação de *ALLOS* com os demais módulos do *SAVAD*. Em seguida, são apresentados os elementos de modelagem do *SAVAD* que permitem modelar uma grande variedade de sistemas de redes de filas, que podem ser solucionados usando a ferramenta *ALLOS*. A interface do *SAVAD* é enfocada posteriormente, juntamente com suas facilidades de interação entre o usuário e o sistema e a flexibilidade para construção e alteração de modelos de redes de filas. Na seção seguinte são mostradas as medidas de desempenho que podem ser obtidas para um modelo solucionado através de *ALLOS*. Finalmente, é apresentado um exemplo que mostra a especificação de um modelo de redes de filas usando *ALLOS*.

2.1 Estrutura do *SAVAD*

Conforme discutido na introdução deste trabalho, a ferramenta *ALLOS* faz parte de um sistema inteligente e integrado para modelagem e avaliação de redes de filas, denominado *SAVAD*. Esta seção focaliza a interação de *ALLOS* com os módulos que compõem o *SAVAD*.

Para melhor compreensão da estrutura de *ALLOS*, apresenta-se, de forma sucinta, a estrutura do *SAVAD*, que pode ser vista na Figura 2.1.

- a) **Interface:** *SAVAD* interage com os usuários através de sua interface, oferecendo-lhes facilidades para a especificação de modelos de redes de filas, a escolha de uma solução para o modelo (opcionalmente) e para a apresentação dos resultados. Outros detalhes sobre a interface são mostrados na seção 2.3.

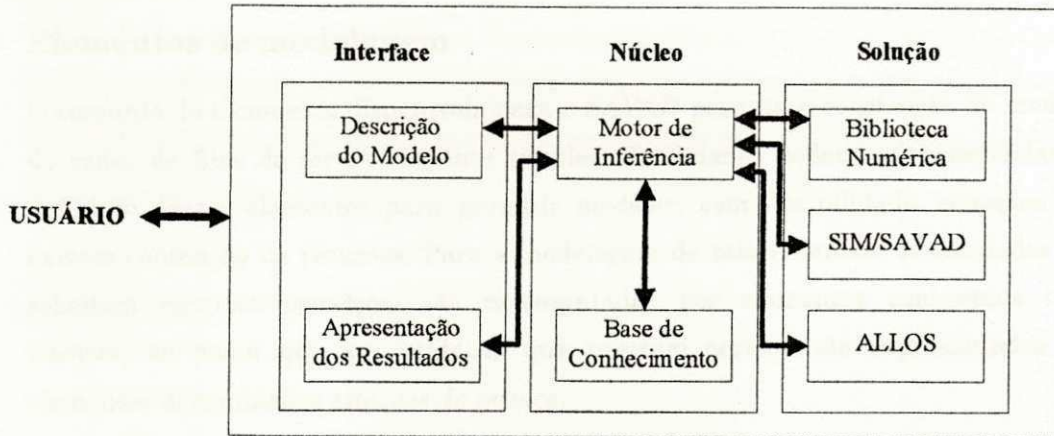


FIGURA 2.1 - Módulos do SAVAD

b) **Núcleo do SAVAD:** O núcleo do sistema é composto de um motor de inferência e de uma base de conhecimento.

- **Base de Conhecimento:** Contém a base de fatos (modelo do usuário) e a base de regras para o reconhecimento dos diversos tipos de redes que podem ser solucionados pelo SAVAD.
- **Motor de Inferência:** Procura conhecer o tipo de rede especificado pelo usuário (base de fatos) a partir das informações disponíveis na base de regras e também qual a solução mais apropriada. Caso o modelo não possua solução analítica conhecida (registrada na base de regras), o modelo será submetido a simulação digital. O usuário também poderá solicitar a simulação do modelo, independentemente deste poder ser solucionado analiticamente. Para maiores detalhes ver [SOUTO, 1993; CABRAL, 1994]

c) **Solução:** O módulo solução é composto por:

- **Biblioteca Numérica:** possui um conjunto de soluções analíticas para modelos de redes de filas.
- **SIM/SAVAD:** Um simulador anexado externamente ao SAVAD, sendo por este acionado, caso não exista solução analítica para o modelo de redes. O usuário também pode, opcionalmente, submeter o modelo ao SIM/SAVAD, mesmo que exista solução analítica para o modelo. Mais detalhes podem obtidos em [CONCEIÇÃO FILHO, 1993]
- **ALLOS:** Ferramenta anexada externamente ao SAVAD, sendo acionada por este para solucionar modelos de redes de filas. Após a especificação de um modelo, por parte do usuário, ALLOS gera automaticamente o espaço de estados do modelo, a partir de um estado inicial e das regras que descrevem o comportamento deste modelo. De posse do espaço de estados do modelo, ALLOS constrói e soluciona a matriz de taxas de transição de estados com a finalidade de obter o vetor de probabilidades de estados e, conseqüentemente, medidas de desempenho relevantes para o modelo.

2.2 Elementos de modelagem

O conjunto de elementos disponíveis para o SAVAD permite a construção de modelos de redes de filas de forma bastante simples. Facilidades podem ser observadas na definição desses elementos para permitir modelar, com flexibilidade, sistemas que exibem contenção de recursos. Para a modelagem de tais sistemas, as entidades que solicitam recursos (serviços) são representadas por elementos conhecidos como *clientes*, ao passo que as entidades que prestam serviço são representadas por elementos denominados *estações de serviço*.

Além desses elementos, na descrição de um modelo faz-se necessário definir os tipos de clientes que existem no modelo e quais os caminhos seguidos pelos clientes. Cada tipo de cliente é representado por um elemento denominado *classe* e os caminhos (seqüências de nodos) são representados por elementos conhecidos como *rotas*.

As disciplinas de atendimento disponíveis para SAVAD são FCFS, LCFS, HOL e aleatória, ao passo que ALLOS utiliza somente a disciplina de atendimento FCFS. SAVAD representa os processos de geração e atendimento de *clientes* utilizando as funções de distribuição de probabilidade exponencial, uniforme, normal, determinística e geral, enquanto que ALLOS utiliza somente a distribuição de probabilidade exponencial.

A seguir, apresenta-se a descrição dos elementos de modelagem disponíveis para SAVAD. Mais detalhes podem ser obtidos em [SOUTO, 1993; CABRAL, 1994].

• Clientes

Clientes são entidades temporárias que circulam através dos elementos (nodos) do modelo, solicitando serviços.

O caminho a ser seguido por um cliente é denominado *rota*. Associada a um cliente existe uma *classe*. As classes permitem que *clientes* com comportamentos diversos possam ser caracterizados e analisados de formas distintas.

Clientes não são especificados explicitamente pelo usuário. Podemos caracterizar como clientes programas submetidos a um computador, mensagens que trafegam em uma rede de computadores, peças em uma linha de montagem, etc.

• Estações de Serviço

Estações de serviço (Figura 2.2) representam os recursos de um modelo de redes de filas. Cada *estação de serviço* pode ser constituída de um ou mais servidores, que são entidades permanentes que atendem os *clientes*, um de cada vez, segundo a função de distribuição de probabilidade exponencial.

Se um *cliente* recém-chegado encontrar a estação ocupada, ele entra na fila de espera. Depois de um certo tempo será selecionado para serviço, de acordo com a disciplina de escalonamento "First Come, First Served" (FCFS).

As filas de espera possuem comprimentos limitados. Caso o usuário especifique um modelo contendo algum elemento de modelagem com comprimento de fila infinito, *ALLOS* comunica ao usuário, que utiliza apenas comprimento de filas limitado e substitui o valor infinito por uma constante para representar esse comprimento.

Denominam-se *servidor simples* e *servidor múltiplo*, respectivamente, aquelas *estações de serviço* que apresentam um servidor e múltiplos servidores.

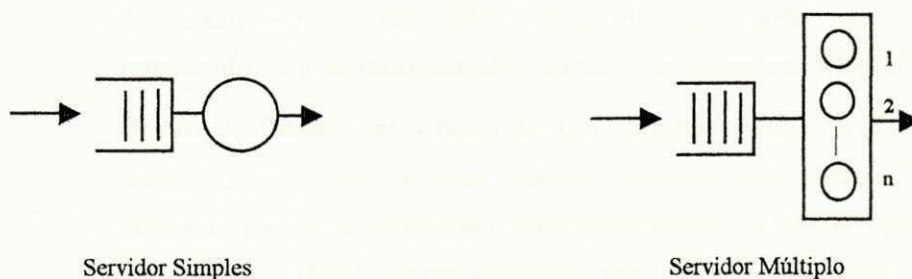


FIGURA 2.2 - Elementos de modelagem do tipo *Estação de serviço*

• Fontes

Fontes (Figura 2.3) são elementos presentes somente em redes abertas (caracterizadas pela topologia onde clientes entram na rede, passam por diversos nodos até serem excluídos ou retirados da rede), representando o processo de chegada de *clientes* no modelo. Os *clientes* são gerados segundo a função de distribuição de probabilidade exponencial, com média fornecida pelo usuário.

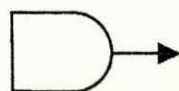


FIGURA 2.3 - Elemento de modelagem do tipo *Fonte*

• Sorvedouros

Sorvedouros (Figura 2.4) são elementos utilizados em redes abertas com a finalidade de eliminar *clientes* do modelo.

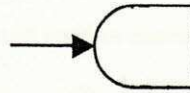


FIGURA 2.4 - Elemento de modelagem do tipo *Sorvedouro*

• **Pontos de Controle**

Pontos de controle (Figura 2.5) são *nodos* que controlam o fluxo de *clientes* em um modelo de redes de filas. Esses elementos foram projetados para fornecer maior flexibilidade na construção de modelos. Os *pontos de controle* disponíveis são o seguintes:

- **Ponto de Multiplicação:** permite a multiplicação de clientes que se destinam a *rotas* diferentes. Este elemento pode ser usado para representar um demultiplexador em modelos de sistemas de difusão.
- **Ponto de Fusão:** faz a fusão de dois ou mais *clientes* de rotas distintas pertencentes a uma mesma *classe*, liberando um único *cliente*. Este elemento possui “n” entradas e uma única saída. Os *clientes* que chegam a um *ponto de fusão* devem aguardar nas diversas entradas para que a fusão ocorra de forma simultânea, gerando um *cliente* na saída.
- **Ponto de Sincronização:** Destina-se a bloquear *clientes* até que determinada condição seja satisfeita. Para tanto, o número de *clientes* em cada uma de suas filas de entrada deve atingir um limite especificado pelo usuário. *ALLOS* permite a especificação do elemento *ponto de sincronização* com uma ou mais entradas/saídas. Entretanto, *SAVAD* apresenta as seguintes configurações:

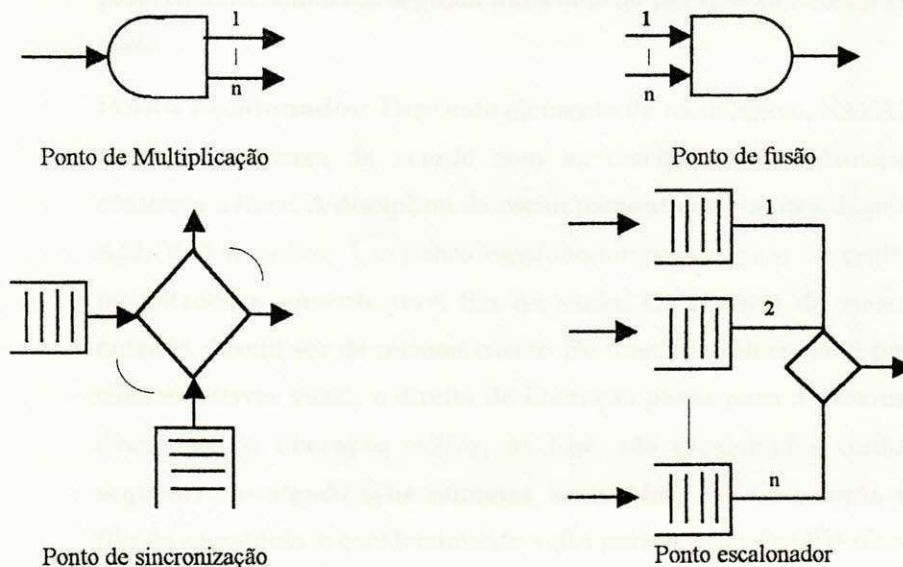


FIGURA 2.5 - Elementos de modelagem do tipo *Ponto de controle*

- a) duas entradas e duas saídas correspondentes;
- b) duas entradas e uma saída associada a uma das entradas;
- c) uma entrada e uma saída correspondente.

Para a configuração (a) a condição de liberação de *clientes* depende das filas de entrada do *ponto de sincronização* alcançarem, ambas, os limites impostos pelo usuário. Esses limites são definidos através dos parâmetros $PS1_i$ ($i = 1, 2$) associados às entradas 1 e 2 do *ponto de sincronização*, respectivamente. A liberação de clientes ocorre quando esses parâmetros de entrada $PS1_i$ alcançam seus limites para, em seguida, serem associados aos parâmetros $PS2_i$ ($i = 1, 2$) de saída do *ponto de sincronização*.

Para a configuração (b), um *ponto de sincronização* com duas entradas e apenas uma saída, os parâmetros $PS1_1$ e $PS1_2$ são associados às filas 1 e 2 de entrada, respectivamente. No caso da fila 1 pertencer a uma rota fechada e a fila 2 a uma *rota* aberta, então $PS1_1$ é associado ao parâmetro $PS2$. No caso das filas 1 e 2 pertencerem a *rotas* abertas, então o valor associado ao parâmetro de saída $PS2$ é escolhido como o menor valor entre $PS1_1$ e $PS1_2$. Para o caso em que ambas as filas de entrada pertencerem a *rotas* fechadas, se $PS1_1$ for igual a $PS1_2$, o valor de $PS1_1$ ($PS1_2$) é associado a $PS2$, contudo se $PS1_1$ for diferente de $PS1_2$, a interface deve rejeitar a especificação do modelo, avisando ao usuário, pois neste caso a população de uma das rotas fechadas poderia ser alterada.

Para um *ponto de sincronização* com apenas uma entrada e uma saída correspondente (configuração (c)), a condição de liberação de *clientes* ocorre quando a fila de entrada alcança o valor do parâmetro $PS1$ definido pelo usuário, sendo em seguida associado ao parâmetro $PS2$ de saída deste *nodo*.

- **Ponto Escalonador:** Para este elemento de modelagem, SAVAD permite escalonar clientes de acordo com as disciplinas de liberação *cíclica*, *aleatória* e *livre*. A disciplina de escalonamento de clientes disponível para ALLOS é a *cíclica*. Um *ponto escalonador* possui uma ou múltiplas filas na entrada e somente uma fila na saída. Os clientes de mesma fila de entrada devem ser de mesma *classe*. Se uma fila selecionada para liberar *clientes* estiver vazia, o direito de liberação passa para a próxima fila. Na disciplina de liberação *cíclica*, as filas são escalonadas conforme uma sequência, contendo seus números associados. Após a seleção da última fila da sequência, o escalonamento volta para a primeira fila da sequência. A disciplina de liberação de *clientes* nas filas pode ser *não-exaustiva*

(somente um *cliente* é liberado em cada fila), *limitada* (até “n” *clientes* são liberados em cada fila) e *exaustiva* (todos os *clientes* da fila são liberados). O número de *rotas* associadas a um *ponto escalonador* é igual ao seu número de filas de entrada.

- **Classes**

As classes são usadas para determinar tipos diferentes de *clientes* em circulação no modelo. *Clientes* de uma mesma classe possuem os mesmos atributos e têm prioridade igual. Um exemplo de classes diferentes em um mesmo sistema é uma rede de computadores com pacotes de dados e pacotes de voz.

- **Rotas**

As rotas estabelecem os diversos caminhos existentes na rede por onde os *clientes* circulam. O roteamento da ferramenta *ALLOS* é fixo. *Rotas* podem ser abertas ou fechadas. *Rotas* abertas necessitam de *nodos* que gerem e, eventualmente, de *nodos* que eliminem *clientes*. *Rotas* fechadas possuem um número fixo de *clientes* (*população*) que circula pela sequência pré-estabelecida dos *nodos*.

Nos modelos a serem solucionados por *ALLOS*, *clientes* podem ficar bloqueados em um nodo quando o próximo nodo, com fila, não dispõe de espaço de armazenamento para recebê-los.

2.3 Interface do SAVAD

A interface do *SAVAD* é constituída por janelas, “menu principal” e “sub-menus” verticais (“top-down”) com a finalidade de auxiliar o usuário na construção de modelos, de forma bastante simples, permitindo uma interação amigável entre o usuário e o sistema.

Inicialmente, o usuário fornece o nome do modelo que deseja especificar. Caso o modelo já exista, ele pode ser facilmente alterado. Após fornecer o nome do modelo, a interface interage com o usuário, solicitando-lhe os elementos de modelagem (descritos na seção 2.2 desta dissertação) que serão utilizados, com seus respectivos atributos. Se algum desses elementos de modelagem já estiver definido, a interface apresenta-o com os seus respectivos parâmetros.

Os componentes utilizados na descrição dos modelos de redes de filas apresentam grande flexibilidade, possibilitando a modelagem de uma grande variedade de sistemas que apresentam contenção de recursos como, por exemplo, sistemas de computação e redes de computadores, sistemas de manufatura, controle de processos e

sistemas de tráfego, além de outros que podem ser modelados utilizando o paradigma de redes de filas.

Na descrição de um modelo, a interface verifica a consistência de cada elemento descrito individualmente e do modelo globalmente, antes de submetê-lo a uma avaliação de desempenho. A interface ainda oferece facilidades, tais como manipulação de arquivos e acesso direto aos comandos do sistema operacional MS-DOS.

Após validar o modelo globalmente, a interface procura uma solução para o modelo e fornece as medidas de desempenho obtidas. Neste caso, o sistema deve escolher adequadamente as técnicas utilizadas, sejam elas exatas (baseadas na Teoria das Filas [KLEINROCK, 1975]) ou aproximadas (baseadas em Simulação Digital [CONCEIÇÃO FILHO, 1993]).

A Figura 2.6 apresenta o menu principal, a janela principal e o menu auxiliar da interface de SAVAD.

O menu principal permanece sempre visível para o usuário, fornecendo-lhe as funções básicas, que são:

- **Arquivo:** Permite a manipulação de arquivos usados para a descrição do modelo. Esta opção possibilita o acesso a um arquivo já existente, criar um novo arquivo, gravar, eliminar, copiar ou renomear arquivos, além de exibir o conteúdo de diretórios.
- **Edição:** Corresponde ao processamento de definição de um modelo através da especificação dos seus elementos de modelagem. Esta opção permite a inserção, alteração, exclusão e cópia dos elementos do modelo, além de permitir a modificação desses elementos.
- **Consulta:** Permite a consulta da configuração do modelo, do resultado da verificação de integridade e das medidas de desempenho obtidas após o processamento de um modelo. A consulta pode ser feita por elemento, categoria de elementos ou do modelo global.
- **Verificação:** Testa a integridade de um modelo especificado pelo usuário. Esta verificação também é feita automaticamente antes de solucionar qualquer modelo especificado ou alterado, evitando a submissão de modelos inconsistentes ou incompletos para solução.
- **Processa:** Faz a verificação de um modelo, caso não tenha sido solicitada pelo usuário e, se este foi especificado corretamente, ativa o seu processamento. É nesta opção que a solução "Cadeias de Markov" pode ser selecionada, fornecendo aos usuários o espaço de estados, o vetor de probabilidades e as medidas de desempenho para o modelo em análise.

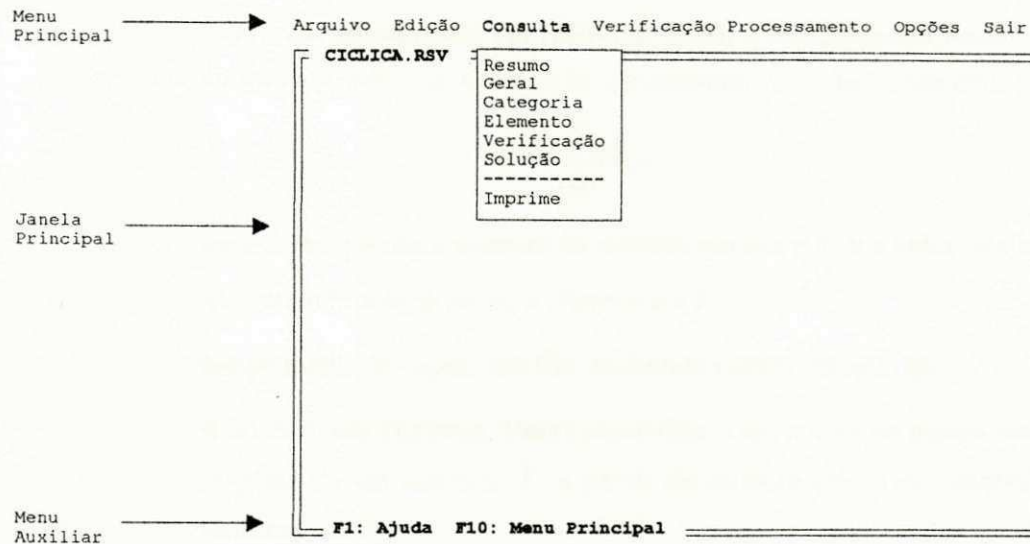


FIGURA 2.6 - Interface básica de SAVAD

- **Opções:** Permite ao usuário o acesso direto ao sistema operacional MS-DOS para a execução de comandos.
- **Sair:** Salva as alterações eventualmente feitas para um modelo e encerra o processamento, retornando ao sistema operacional.

2.4 Medidas de desempenho

A avaliação da capacidade de um sistema para suportar a demanda de utilização é essencial para sua verificação, desenvolvimento e uso adequados [MOURA, 1986]. Normalmente, a análise de desempenho é feita observando-se o comportamento de algumas medidas de interesse com as variações da carga do sistema. É importante, porém, que seja feita a nível de projeto para garantir a especificação e o uso adequado, além de reduzir os custos totais [SOUTO, 1993].

Como resultado do processamento através de cadeias de Markov, ALLOS gera um arquivo com o vetor de probabilidades e as medidas de desempenho, que são exibidos na janela principal da interface. As medidas de desempenho também ficam disponíveis para o usuário através da opção *consulta* do menu principal desta interface.

Seguem essas medidas de desempenho, conforme estejam associadas a uma das seguintes categorias: *nodos* e *rotas*.

- **Nodos**

⇒ *Estação de serviço*

— comprimento médio de fila:

Conforme apresentado em [MOURA, 1986] o comprimento médio de fila ou número médio de clientes de um sistema, $\bar{\mathcal{N}}$, é dado pela equação

$$\bar{\mathcal{N}} = \sum_{k=0}^{\infty} k \mathcal{P}_k,$$

onde k representa o número de clientes em fila e \mathcal{P}_k é o vetor que contém as probabilidades de se ter k clientes em fila.

- tempo médio de espera em fila, incluindo cliente em serviço:

A lei de Little [LITTLE, 1961] possibilita a derivação do tempo médio de resposta de um sistema, \bar{T} , a partir do número médio de clientes neste sistema, isto é,

$$\bar{T} = \frac{\bar{\mathcal{N}}}{\lambda},$$

onde λ representa a taxa média de chegada de clientes ao sistema.

- fator de utilização:

Para um modelo de redes de filas com n filas, tem-se o vetor $\mathcal{P}(\vec{\mathcal{K}})$, em que cada componente de $\vec{\mathcal{K}} = (\mathcal{K}_1, \dots, \mathcal{K}_n)$ representa o número de clientes no nó i e $\mathcal{P}(\vec{\mathcal{K}})$ é o vetor que contém a probabilidade associada a cada estado da cadeia. Suponha-se que o espaço de estados para este modelo contenha m estados. Então, o fator de utilização para cada nó i é dado pela expressão [SAUER, 1981]:

$$\rho_i = \sum_{j=1}^m \mathcal{P}(\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_i, \dots, \mathcal{K}_n),$$

onde $\mathcal{K}_i > 0$ e $1 \leq i \leq n$.

⇒ Ponto de Sincronização

- comprimento médio em cada fila:

$$\bar{\mathcal{N}} = \sum_{k=0}^{\mathcal{K}} k \mathcal{P}_k$$

onde \mathcal{K} representa o comprimento máximo de fila.

- tempo médio de espera em cada fila:

$$\bar{T} = \frac{\bar{\mathcal{N}}}{\lambda}$$

⇒ Ponto Escalonador

- comprimento médio em cada fila:

$$\bar{\mathcal{N}} = \sum_{k=0}^{\mathcal{K}} k P_k$$

onde \mathcal{K} representa o comprimento máximo de fila.

— tempo médio de espera em cada fila:

$$\bar{T} = \frac{\bar{\mathcal{N}}}{\lambda}$$

• Rotas

Seja \mathcal{R} o número de rotas do modelo.

— vazão média para rotas abertas: Seja uma rede de filas com n nós em série, sem realimentação. Os tempos de interchegada de clientes para o nó 1 são distribuídos exponencialmente com média $1/\lambda$ segundos. Os tempos de serviço do servidor i , $i = 1, \dots, n$ têm distribuição exponencial com média $1/\mu_i$ segundos. De acordo com [BURKE, 1956], em regime estacionário a saída de um sistema M/M/m com taxa de entrada λ e cada servidor operando com taxa μ é um processo de Poisson com taxa λ , estatisticamente independente do processo de entrada. Como consequência do Teorema de Burke, cada nó i ($i > 1$) é um sistema M/M/m, com taxa de chegada de clientes λ e pode ser analisado independentemente do nó 1.

Para um modelo de redes de filas com n filas, tem-se o vetor $\vec{P}(\mathcal{K})$, em que cada componente de $\vec{\mathcal{K}} = (\mathcal{K}_1, \dots, \mathcal{K}_n)$ representa o número de clientes no nó i e $\vec{P}(\mathcal{K})$ é o vetor que contém a probabilidade de cada estado da cadeia.

A vazão média para uma rota pertencente a uma rede aberta com comprimento de fila limitado é dado pela equação [CABRAL, 1987]:

$$\gamma_r = \lambda_r [1 - \sum \mathcal{P}(\mathcal{K}_1, \mathcal{K}_2, \dots, Max, \dots, \mathcal{K}_n)], r = 1, \dots, \mathcal{R}$$

onde λ_r é a taxa de interchegadas de clientes e Max é o comprimento máximo da fila conectada à fonte, associada à rota r .

— vazão média para rotas fechadas: conforme discutido em [MOURA, 1986], em uma rede de filas com n nós, em que o número de clientes é constante e finito e não existem chegadas externas, a taxa de chegada para o nó i , λ_i , é dada pela equação:

$$\lambda_i = \sum_{j=1}^n \lambda_j d_{ji}, i = 1, 2, \dots, n; i \in r \text{ e } r = 1, \dots, \mathcal{R}$$

onde d_{ji} é a probabilidade de um cliente, ao sair do nó j , se destine ao nó i . Como a fórmula acima gera n equações linearmente dependentes, o valor de λ_i pode ser encontrado substituindo-se uma das equações por

$$\sum_{j=1}^n d_{ji} = 1$$

— atraso médio fim-a-fim \mathcal{T}_r da rota r :

$$\mathcal{T}_r = \sum_{i=1}^n t_i, \text{ se } i \in r \text{ e } r = 1, \dots, \mathcal{R}$$

onde t_i é o tempo de resposta associado ao nodo i e \mathcal{R} é o número de nodos que possuem filas.

2.5 Exemplo de especificação de um modelo

A Figura 2.7 representa uma rede cíclica contendo uma UCP e dois dispositivos idênticos de entrada/saída (E/S) conforme discutida em [SAUER, 1981]. A disciplina de escalonamento utilizada é FCFS e os tempos de serviço são independentes e identicamente distribuídos exponencialmente.

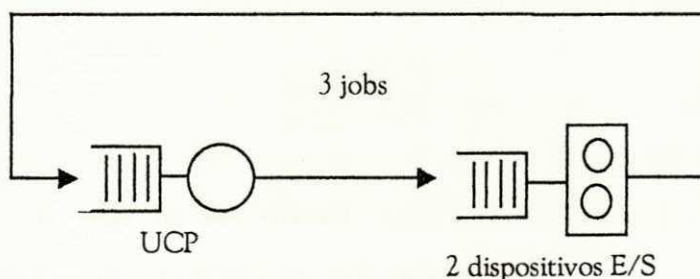


FIGURA 2.7 - Modelo de uma rede cíclica

A especificação para o modelo de uma rede cíclica pode ser vista através das Figuras 2.8, 2.9 e 2.10.

Convém observar que o comprimento de fila especificado como ilimitado para UCP e E/S pode ser usado sem restrição nesse caso, uma vez que para cadeias fechadas, ALLOS atribui o valor da população ao comprimento máximo de cada fila que compõe a cadeia.

A partir da geração automática do espaço de estados que comporá a matriz de transição do modelo, a ferramenta ALLOS soluciona o modelo usando o método apresentado em [SAUER, 1981]. Outros métodos para a solução da matriz de transição de estados podem ser encontrados em [SILVA, 1992b].

```

Arquivo Edição Consulta Verificação Processamento Opções Sair
CICLICA.RSV - INSERÇÃO DE ELEMENTOS

TIPO.....: Estação de Serviço
NOME.....: UCP
TIPO DE SERVIDOR.....: Simples
COMPRIMENTO DA FILA..: Ilimitado
QUANT. DE SERVIDORES.: Único
DISCIPLINA.....: FCFS
DISTRIBUIÇÃO:
    TIPO.....: Exponencial
    MÉDIA.....: 6.67ms

F1: Ajuda F10: Menu Principal
    
```

FIGURA 2.8 - Especificação do elemento de modelagem UCP

```

Arquivo Edição Consulta Verificação Processamento Opções Sair
CICLICA.RSV - INSERÇÃO DE ELEMENTOS

TIPO.....: Estação de Serviço
NOME.....: E/S
TIPO DE SERVIDOR.....: Múltiplo
COMPRIMENTO DA FILA..: Ilimitado
QUANT. DE SERVIDORES.: 2
DISCIPLINA.....: FCFS
DISTRIBUIÇÃO:
    TIPO.....: Exponencial
    MÉDIA.....: 10.0ms

F1: Ajuda F10: Menu Principal
    
```

FIGURA 2.9 - Especificação do elemento de modelagem E/S

```

Arquivo Edição Consulta Verificação Processamento Opções Sair
CICLICA.RSV - INSERÇÃO DE ELEMENTOS

TIPO.....: Rota
NOME.....: rotal
CLASSE.....: Classel

[ ROTA ]
UCP -> E/S -> UCP

TIPO.....: Fechada
POPULAÇÃO.....: 3

F1: Ajuda F10: Menu Principal
    
```

FIGURA 2.10 - Especificação do elemento de modelagem Rota

Após solucionar o modelo, *ALLOS* apresenta as medidas de desempenho, através da interface do *SAVAD*, e o vetor de probabilidades do modelo. Pode-se verificar estes resultados através das Figuras 2.11 e 2.12, respectivamente. Na Figura 2.13 mostra-se o diagrama de transição de estados para o modelo de uma rede cíclica, conforme discutido em [SAUER, 1981].

```

Arquivo Edição Consulta Verificação Processamento Opções Sair
CICLICA.RSV - SOLUÇÃO

TIPO DE MODELO DE REDE...: Rede Fechada
SOLUÇÃO.....: Cadeias de Markov

MEDIDAS DE DESEMPENHO:

FATOR DE UTILIZAÇÃO DA UCP.....: 0.812
FATOR DE UTILIZAÇÃO DE CADA DISPOSITIVO DE E/S: 0.609
VAZÃO DA UCP.....: 0.122
COMPRIMENTO MÉDIO DE FILA DA UCP.....: 1.596

Enter: Continua
    
```

FIGURA 2.11 - Medidas de desempenho para o modelo de uma rede cíclica

```

p[3,0]= 0.224409  p[2,1]= 0.335901  p[1,2]= 0.251426  p[0,3]= 0.188265
    
```

FIGURA 2.12 - Vetor de probabilidades para o modelo de uma rede cíclica

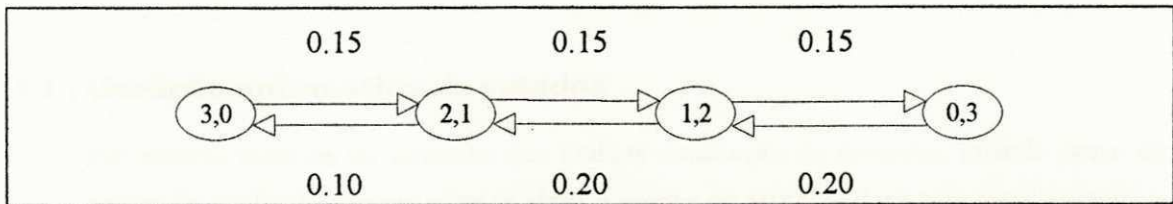


FIGURA 2.13 - Diagrama de transições entre estados para o modelo de uma rede cíclica

No capítulo 3 serão apresentados o processo de geração automática, os procedimentos para se chegar à matriz de transição de estados e o método de solução considerado.

3 *ALLOS* - Concepção e desenvolvimento

Modelos de sistemas reais normalmente dão origem a uma cadeia de Markov, com um número elevado de estados. Um problema não menos importante diz respeito à geração automática de estados e taxas (ou probabilidades) de transição entre os estados gerados [SILVA, 1992b]. É importante que o usuário possa especificar o modelo a partir de uma ferramenta amigável, e que as medidas de desempenho de interesse possam também ser observadas de forma automática, sem que o usuário necessite conhecer os detalhes da representação matemática.

Este capítulo focaliza o processo de geração automática de estados, os procedimentos para a obtenção do espaço de estados associado a um modelo, juntamente com a matriz de taxas de transição entre os estados, e o método de solução adotado para *ALLOS*, para chegar ao vetor de probabilidades associado aos estados da cadeia. De posse do vetor de probabilidades, a ferramenta apresenta as medidas de desempenho relevantes para o modelo, conforme seção 2.4. No final do capítulo são apresentadas a estrutura básica da ferramenta *ALLOS* e sua interação com os demais módulos do *SAVAD*.

3.1 Geração automática de estados

Em muitos modelos de sistemas que exibem contenção de recursos, grande parte do tempo de avaliação é gasto para se obter o espaço de estados. Para tais modelos, várias medidas de interesse podem ser calculadas a partir de um número relativamente pequeno de estados.

O maior problema, na solução de modelos através de cadeias de Markov, é a grande cardinalidade do espaço de estados. Um sistema com N componentes, cada um podendo estar em K estados diferentes, possui um espaço de estados de dimensão K^N . Tomando-se, por exemplo, $K = 10$ e $N = 5$, obtém-se um espaço de estados de 10^5 . Desta forma, para cada componente adicional o espaço de estados cresce com um fator de 10. Entretanto, os sistemas de computação continuam a crescer tanto em velocidade

de acesso à memória quanto em velocidade de processamento, tornando possível, atualmente, a solução de modelos com espaço de estados na ordem de 10^5 a 10^6 [SILVA, 1992b].

A obtenção do espaço de estados associado a um modelo ainda não é uma tarefa fácil. Uma das alternativas disponíveis na literatura é a exploração dinâmica do espaço de estados [SILVA, 1992a; SILVA, 1992b, DINIZ, 1991]. A idéia básica de técnicas de exploração dinâmica é a de desenvolver algoritmos para guiar o processo de geração da matriz de transição de estados, cuja importância de um estado deve ser dada em termos da sua contribuição para o cálculo das medidas de desempenho de interesse.

Uma forma elegante e eficiente para a exploração dinâmica é a geração automática de estados, em que, a partir da especificação do modelo de um sistema por parte do usuário, a matriz de transição de estados é gerada a partir de um estado inicial e das regras que descrevem o comportamento do sistema.

O método de exploração dinâmica discutido em [DIMITRIJEVIC, 1989;1990] é baseado no cálculo recursivo do número esperado de visitas a um estado entre visitas consecutivas a um dado estado inicial. Existem dois conjuntos de estados neste procedimento de geração dinâmica. O primeiro é chamado de conjunto explorado, $S_e^{(n)}$, e contém os estados que serão incluídos no modelo final e, o segundo, conjunto não explorado, $S_u^{(n)}$, contém os estados vizinhos daqueles em $S_e^{(n)}$. O critério de parada deve ser relacionado com o tempo de permanência nos estados gerados.

No caso de *ALLOS*, o critério de parada usado no processo de geração de estados está relacionado com o comprimento de filas do modelo em análise, visto que *ALLOS* utiliza somente sistemas de filas M/M/m [KLEINROCK, 1975]. Neste caso, a explosão de estados é finalizada quando o comprimento de todas as filas do modelo alcança o seu limite.

Em [DINIZ, 1990] foi proposta uma ferramenta que implementa o paradigma de modelagem orientado a objeto, permitindo a descrição de modelos em alto nível. Seu domínio de aplicação abrange análise de desempenho e confiabilidade de sistemas. No entanto, esta ferramenta, apesar de gerar a matriz de taxas de transição, não obtém o vetor de probabilidades associado aos estados do modelo e conseqüentemente não apresenta as medidas de desempenho. Quanto a *ALLOS*, esta restringe seu domínio de aplicação para a modelagem e avaliação de desempenho de sistemas, supondo-se que estes são perfeitamente confiáveis. Entretanto, *ALLOS* possui certas vantagens, uma vez que soluciona os modelos markovianos e apresenta as medidas de desempenho mais relevantes.

3.2 Procedimentos de geração automática de estados

O esquema apresentado na Figura 3.1 representa as possíveis transições de estados dos modelos que utilizam os elementos de modelagem apresentados na seção 2.2.

Na convenção adotada, cada elemento de modelagem pode ser precedido e seguido por uma ou mais filas. Exceções se aplicam aos elementos fonte (não pode ser precedido por fila) e sorvedouro (não pode ser seguido por fila). A notação usada nesses casos "[]", representa uma fila nula.



FIGURA 3.1 - Possíveis transições entre os estados dos modelos

A partir de um estado (k_1, \dots, k_n) , onde k_i é o número de clientes na fila i e n é o número de filas do modelo, o conjunto de estados alcançáveis a partir desse estado é definido conforme as seguintes regras:

1. [] fonte [f_j]
 Se $k_{f_j} < \text{Max}_{f_j}$
 Então $(k_{f_1}, \dots, k_{f_j}, \dots, k_{f_n}) \Rightarrow_1 (k_{f_1}, \dots, k_{f_j}+1, \dots, k_{f_n})$
2. [f_i] sorvedouro []
 Se $k_{f_i} > 0$
 Então $(k_{f_1}, \dots, k_{f_i}, \dots, k_{f_n}) \Rightarrow_2 (k_{f_1}, \dots, k_{f_i}-1, \dots, k_{f_n})$
3. [f_i] servidor [f_j]
 Se $(k_{f_i} > 0) \wedge (k_{f_j} < \text{Max}_{f_j})$
 Então $(k_{f_1}, \dots, k_{f_i}, \dots, k_{f_j}, \dots, k_{f_n}) \Rightarrow_3 (k_{f_1}, \dots, k_{f_i}-1, \dots, k_{f_j}+1, \dots, k_{f_n})$
4. [f_j] multiplicação [f_{j₁}, ..., f_{j_m}]
 Se $(k_{f_j} > 0) \wedge (\forall \alpha \mid 1 \leq \alpha \leq m, k_{f_{j_\alpha}} < \text{Max}_{f_{j_\alpha}})$
 Então $(k_{f_1}, \dots, k_{f_i}, \dots, k_{f_j}, \dots, k_{f_n}) \Rightarrow_4 (k_{f_1}, \dots, k_{f_i}-1, \dots, k_{f_{j_\alpha}}+1, \dots, k_{f_n}), \forall \alpha \mid 1 \leq \alpha \leq m$

5. $[f_{i_1}, \dots, f_{i_m}]$ escalonador $[f_j]$

$$\forall \alpha \mid 1 \leq \alpha \leq m$$

$$\text{Se } (k_{f_{i_\alpha}} \geq \text{PE}_{f_{i_\alpha}}) \wedge (k_{f_j} + \text{PE}_{f_{i_\alpha}} \leq \text{Max}_{f_j})$$

$$\text{Então } (k_{f_1}, \dots, k_{f_{i_\alpha}}, \dots, k_{f_j}, \dots, k_{f_n}) \Rightarrow (k_{f_1}, \dots, k_{f_{i_\alpha}} - \text{PE}_{f_{i_\alpha}}, \dots, k_{f_j} + \text{PE}_{f_{i_\alpha}}, \dots, k_{f_n}), \forall \alpha \mid 1 \leq \alpha \leq m$$

6. $[f_{i_1}, \dots, f_{i_m}]$ sincronizador $[f_j, \dots, f_{j_p}]$

$$\text{Se } (\forall \alpha \mid 1 \leq \alpha \leq m, k_{f_{i_\alpha}} \geq \text{PS1}_{f_{i_\alpha}}) \wedge (\forall \beta \mid 1 \leq \beta \leq p, k_{f_{j_\beta}} + \text{PS2}_{f_{j_\beta}} \leq \text{Max}_{f_{j_\beta}})$$

$$\text{Então } (k_{f_1}, \dots, k_{f_{i_\alpha}}, \dots, k_{f_{j_\beta}}, \dots, k_{f_n}) \Rightarrow (k_{f_1}, \dots, k_{f_{i_\alpha}} - \text{PS1}_{f_{i_\alpha}}, \dots, k_{f_{j_\beta}} + \text{PS2}_{f_{j_\beta}}, \dots, k_{f_n}), \forall \alpha \mid 1 \leq \alpha \leq m, \forall \beta \mid 1 \leq \beta \leq p$$

7. $[f_{i_1}, \dots, f_{i_m}]$ fusão $[f_j]$

$$\text{Se } (\forall \alpha \mid 1 \leq \alpha \leq m, k_{f_{i_\alpha}} \geq 0) \wedge (k_{f_j} < \text{Max}_{f_j})$$

$$\text{Então } (k_{f_1}, \dots, k_{f_{i_\alpha}}, \dots, k_{f_j}, \dots, k_{f_n}) \Rightarrow (k_{f_1}, \dots, k_{f_{i_\alpha}} - 1, \dots, k_{f_j} + 1, \dots, k_{f_n}), \forall \alpha \mid 1 \leq \alpha \leq m$$

onde:

- k_i é o comprimento da fila i ,
- Max_i é o comprimento máximo da fila i ,
- PE_i é o parâmetro associado à liberação de clientes das filas i do ponto escalonador,
- PS1_i é o parâmetro associado à liberação de clientes da fila i do ponto de sincronização,
- PS2_j é o parâmetro associado à liberação de clientes do ponto de sincronização para a fila j ,

e onde $E_a \xrightarrow{i} E_b$, ou simplesmente $E_a \Rightarrow E_b$, representa a existência de uma transição direta, correspondente ao elemento de modelagem i (regra i), entre os estados E_a e E_b .

Sejam E_a e E_b dois estados. Diz-se que E_b é alcançável a partir de E_a ($E_a \xrightarrow{i}^* E_b$) se, e somente se, existe um número finito de transições entre E_a e E_b ($E_a \Rightarrow E_{a+1} \Rightarrow \dots \Rightarrow E_b$). A transição de estado E_a para um E_b , é dita *válida* ($E_a \xrightarrow{val} E_b$) se, e somente se,

$$\nexists E_c \mid E_b \xrightarrow{i}^* E_c, i = 5,6$$

onde $i = 5,6$ representam, respectivamente, as transições associadas aos pontos escalonador e sincronizador.

Assim, a geração de uma transição válida é feita segundo a seguinte regra:

$$\text{Se } \exists E_c \mid ((E_b \xrightarrow{i} E_c, i = 5,6) \wedge \nexists E_d \mid (E_c \xrightarrow{i} E_d, i = 5,6))$$

$$\text{Então } E_a \xrightarrow{val} E_c$$

Senão $E_a \stackrel{val}{\Rightarrow} E_b$, desde que $\nexists E_c \mid E_b \stackrel{*}{\Rightarrow} E_c, i = 5,6$

Para modelos de redes fechadas, a seguinte restrição pode ser anexada à regra anterior:

$$\sum_{i=1,n} k_i = \text{população},$$

em que k_i representa o tamanho da fila i e n o número total de filas do modelo.

Portanto, para uma rede de filas F , um estado inicial E_0 é definido genericamente como:

If F é aberta

Então $E_0 = (0, \dots, 0)$

Senão $E_0 = (P, 0, \dots, 0)$

em que P é a população da cadeia fechada. Caso $P > \kappa_1$, então o excedente $E = (P - \kappa_1)$ passará para κ_2 , e assim sucessivamente, até que $E < \kappa_i$ ($i = 1, \dots, n$).

Finalmente, o procedimento de geração da cadeia de Markov é definido como:

A partir de um conjunto de estados formado de um estado inicial E_0

constroi-estado($[E \mid Es]$) :- **If** E ainda não foi contruído,

Então $X = \{\text{Estado: } E \Rightarrow \text{Estado}\}$

constroi-estado($Es \mid X$)

Senão constroi-estado(Es)

onde $[A]$ representa uma lista com um único elemento A e $A \mid B$ representa a concatenação das listas A e B .

O procedimento, acima, pode ser interpretado informalmente da seguinte maneira:

Seja $EE = \{E_0\}$ o espaço de estados para um modelo de redes de filas. Para cada novo estado gerado E_i

Se $E_i \notin EE$,

então $EE \leftarrow EE \cup \{E_i\}$

3.3 Solução da matriz de taxas de transição de estados

Para um grande número de aplicações é necessário encontrar a solução, em regime estacionário, do sistema sendo analisado. Embora o usuário esteja, muitas vezes, interessado no comportamento do sistema durante um intervalo de tempo finito $(0, t)$ a solução em estado estacionário pode fornecer uma boa aproximação para valores de t suficientemente grandes [SILVA, 1992b]. A partir desta solução as medidas de interesse podem ser calculadas.

Pode-se dividir os métodos usados para solução de cadeias de Markov (ou, em geral, um sistema de equações lineares) em *métodos diretos e métodos iterativos* [BARROSO, 1987]. Um método é chamado direto quando fornece a solução exata desejada, após um número finito de passos, enquanto um método é chamado iterativo quando fornece uma sequência de soluções aproximadas que convergem para o valor correto.

Em geral, métodos diretos são apropriados quando o número de estados do sistema modelado não é muito grande (na ordem de até um ou poucos milhares de estados) e quando a matriz de transição de estados do sistema não é esparsa. Métodos iterativos são adequados quando a matriz de transição de estados é grande e esparsa, pois eles preservam esta última propriedade [SAUER, 1981; SILVA, 1992b].

Geralmente, métodos iterativos envolvem multiplicação de um vetor por uma ou mais matrizes obtidas da matriz original. Um ponto importante a ser considerado para avaliar a eficácia de um método iterativo é a rapidez com que o método converge para a solução final [SILVA, 1992b].

ALLOS utiliza o método de solução iterativa apresentado em [SAUER, 1981]. Inicialmente, assume-se que existe a probabilidade $\mathcal{P}_i(t)$ representando o sistema no estado S_i no tempo t . O equilíbrio do sistema existe, desde que

$$\lim_{t \rightarrow \infty} \mathcal{P}_i(t) = \mathcal{P}_i, \quad 1 \leq i \leq n.$$

Pode-se, ainda, assumir que $\mathcal{P}_i \geq 0$ para $1 \leq i \leq n$. Desde que o equilíbrio exista, espera-se que as taxas de transição de entrada para cada estado S_i seja igual as taxas de transição de saída. As taxas de transição de saída do estado S_i são $\mathcal{P}_i a_i$, ao passo que as taxas de transição de entrada no estado S_i são

$$\sum_j \mathcal{P}_j a_j q_{ji}.$$

Portanto, tem-se n equações da forma

$$\mathcal{P}_i a_i = \sum_j \mathcal{P}_j a_j q_{ji}, \quad 1 \leq i \leq n.$$

Essas equações podem ser reescritas como

$$-\mathcal{P}_i a_i + \sum_j \mathcal{P}_j a_j q_{ji} = 0, \quad 1 \leq i \leq n. \quad (1)$$

Pode-se perceber que a n -ésima equação é redundante, ou seja, a partir de qualquer das $n-1$ das equações, pode-se derivar a equação restante. É sabido que

$$\sum_i \mathcal{P}_i = 1.$$

Portanto, pode-se resolver o conjunto de equações lineares

$$-\mathcal{P}_i a_i + \sum_j \mathcal{P}_j a_j q_{ji} = 0, \quad 1 \leq i \leq n-1.$$

$$\sum_i \mathcal{P}_i = 1$$

para obter \mathcal{P}_i , $1 \leq i \leq n$. A partir dessas probabilidades de equilíbrio, as medidas de desempenho apresentadas na seção 2.4 são obtidas.

Desta forma, métodos diretos para solucionar as equações de balanço, tal como o de eliminação de Gauss, será inapropriado, pois este método muda os elementos da matriz, fazendo com que muitos elementos nulos se tornem não nulos. Neste caso, o aumento na capacidade de armazenamento será proibitivo [SAUER, 1981].

Com base no exposto, pode-se concentrar a atenção em soluções iterativas e algoritmos recursivos especializados.

Seja S a matriz de transição de estados de um modelo markoviano, tal que

$$S_{ij} = \begin{cases} a_j q_{ji}, & \text{se } i \neq j \\ -a_i(1 - q_{ii}), & \text{se } i = j \end{cases} \quad (2)$$

Pode-se perceber, neste caso, que S é a matriz correspondente ao conjunto redundante de equações (1), sem a substituição de qualquer uma das equações por $\sum p_i = 1$. Assim, $S.p = 0$, onde p é o vetor coluna $(p_1, p_2, \dots, p_n)^T$ e 0 é o vetor coluna, em que todos os seus elementos são nulos. Portanto, pode-se ter $\Delta S.p + p = p$, de forma que $(\Delta S + I).p = p$, onde I é a matriz identidade.

A última equação sugere, então, a fórmula iterativa

$$\hat{p}^{k+1} = (\Delta S + I)\hat{p}^k$$

onde \hat{p}^k é a estimativa de p na k -ésima iteração. Pode ser mostrado ainda, que através de uma escolha apropriada de Δ , esta iteração convergirá para p com qualquer estimativa de \hat{p}^0 . Conforme discutido em [WALLACE, 1966], um valor de Δ comumente utilizado é

$$\frac{0.99}{\max_i |S_{ii}|}$$

Nota-se que a iteração mantém o valor de $\Delta S + I$ inalterado.

Considerando-se o modelo da rede cíclica apresentada na seção 2.5, tem-se o seguinte diagrama de transição de estados, apresentado na Figura 3.2.

Para este modelo, dada a transição do estado (1,2) para o estado (0,3), tem-se, pela equação (2):

$$(0.15+0.20) \frac{0.15}{0.15+0.20} = 0.15$$

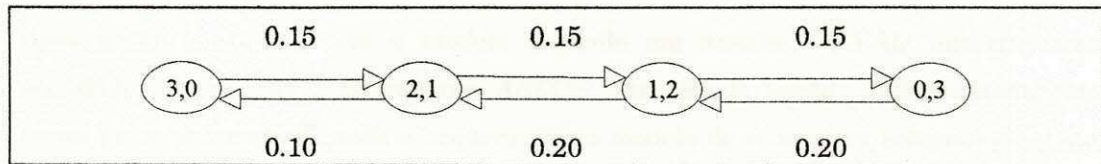


FIGURA 3.2 - Diagrama de transições entre estados para o modelo de uma rede cíclica

Por analogia, pode-se obter todas as taxas de transição entre os estados para o modelo da rede cíclica. A matriz S_{ij} para este modelo é a seguinte:

$$S = \begin{bmatrix} -0.15 & 0.10 & 0 & 0 \\ 0.15 & -0.25 & 0.20 & 0 \\ 0 & 0.15 & -0.35 & 0.20 \\ 0 & 0 & 0.15 & -0.20 \end{bmatrix}$$

Após 17 iterações obtém-se o vetor P_i , onde $P_1 = 0.2244$, $P_2 = 0.3359$, $P_3 = 0.2514$ e $P_4 = 0.1882$, conforme mostrado em [SAUER, 1981].

3.4 Estrutura de ALLOS

Conforme apresentado na Figura 2.1, ALLOS está anexada ao módulo solução do SAVAD, através da opção Cadeias de Markov. Descrevem-se, agora, os detalhes da estrutura de ALLOS e sua interação com os demais módulos do SAVAD, com base na Figura 3.3.

ALLOS é constituída, basicamente, de três módulos: *adaptação*, *geração* e *solução*.

Inicialmente, o usuário faz a especificação de um modelo de redes de filas, através da interface, conforme apresentado na seção 2.5. Os parâmetros do modelo são armazenados em um arquivo. No exemplo do modelo da rede cíclica, os parâmetros gerados através da especificação do modelo são armazenados no arquivo *ciclica.rsv*.

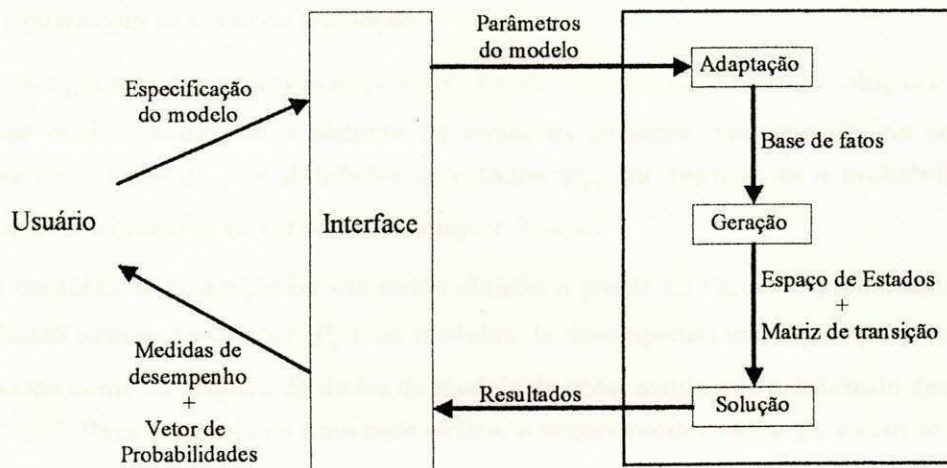


FIGURA 3.3 - Estrutura de ALLOS no ambiente SAVAD

Após gerar o arquivo com o modelo de rede em análise, *SAVAD* encerra suas atividades, acionando a ferramenta *ALLOS*, através da opção "processamento" do menu principal, especificando o arquivo com o modelo de rede a ser solucionado. Este processo ocorre de forma transparente ao usuário, que apenas recebe um aviso de que a solução por Cadeias de Markov foi iniciada. Desta forma, toda memória do computador fica disponível para *ALLOS*.

O módulo Adaptação de *ALLOS* faz a adequação dos parâmetros do modelo (gerados pela interface), constituindo na base de fatos para o módulo Geração.

A Figura 3.4 mostra o arquivo com a base de fatos para o modelo de uma rede cíclica, apresentado na seção 2.5.

```

num_filas(2).
comprimento_fila(1,3).
comprimento_fila(2,3).
restricao([1,2], '>=', 0).
restricao([1,2], '=', 3).
transicao([1], serv, [2]).
transicao([2], serv, [1]).
taxa(1,2,0.15).
taxa(2,1,0.10).
serv_multiplo([[1,1],[2,2]]).

```

FIGURA 3.4 - Arquivo com a base de fatos para o modelo de uma rede cíclica

De posse da base de fatos, o módulo Geração inicia o processo de geração automática de estados para o modelo em análise. A partir de um estado inicial e das regras que descrevem o comportamento do modelo, esse módulo gera dois arquivos. O primeiro contém o espaço de estados (*espest.mrk*) e o segundo contém a matriz de taxas de transição entre os estados do modelo (*transic.mrk*). A Figura 3.5 apresenta o arquivo contendo o espaço de estados para o modelo de uma rede cíclica e a Figura 3.6, o arquivo com as taxas de transição.

Após a geração dos arquivos *espest.mrk* e *transic.mrk*, o módulo solução é ativado. Esse módulo soluciona o sistema de equações lineares, apresentado na seção 3.3, obtendo o vetor de probabilidades de estados \mathcal{P}_i , que representa a probabilidade da cadeia se encontrar no estado i , no tempo t ($t \rightarrow \infty$).

As medidas de desempenho são então obtidas a partir do vetor de probabilidades \mathcal{P}_i . *ALLOS* armazena o vetor \mathcal{P}_i e as medidas de desempenho em um arquivo que tem o mesmo nome do arquivo de dados do modelo de rede, sendo que a extensão deste agora é ".ssv". Para o modelo de uma rede cíclica, o arquivo com o vetor \mathcal{P}_i e com as medidas de desempenho do modelo seria *ciclica.ssv*.

Nº	Estado
1	[3,0]
2	[2,1]
3	[1,2]
4	[0,3]

FIGURA 3.5 - Arquivo `espest.mrk` para o modelo da rede cíclica

Estado <i>i</i>	Estado <i>j</i>	Taxa
1	2	0.15
2	1	0.10
2	3	0.15
3	2	0.20
3	4	0.15
4	3	0.20

FIGURA 3.6 - Arquivo `transic.mrk` para o modelo da rede cíclica

Quando o arquivo com os resultados é gerado a interface lê esse arquivo e apresenta ao usuário as medidas de desempenho mais relevantes do modelo, juntamente com o vetor de probabilidades \mathcal{P}_i .

Convém ressaltar que o processo compreendido entre a adaptação dos parâmetros do modelo até a obtenção das medidas de desempenho é totalmente transparente para o usuário, ou seja, o usuário não percebe a representação matemática do modelo. Esta é uma das grandes potencialidades de *ALLOS*, não exigindo do usuário o conhecimento de um especialista em avaliação de desempenho.

3.5 Desenvolvimento de *ALLOS*

ALLOS foi desenvolvida em ambiente MS-DOS. O módulo de geração automática também está disponível para ambiente Unix.

A interface de *ALLOS* é a mesma definida para o *SAVAD*. A linguagem de programação selecionada para seu desenvolvimento foi C++, versão 2.1 da ZORTECH [ZORTECH INC., 1990], pelo fato desta linguagem suportar o paradigma de objetos, que tem por objetivos aumentar a produtividade do programador através de maior expansibilidade e a reutilização de software, além de controlar a complexidade e o custo da manutenção do mesmo [WIENER, 1991]. Os detalhes sobre a implementação da interface do *SAVAD* podem ser obtidos em [SOUTO, 1993].

O módulo adaptação, que transforma os parâmetros do modelo, especificado pelos usuários, na base de fatos para o módulo geração automática, foi desenvolvido na linguagem de programação C++, versão 2.1 da ZORTECH, sendo constituído de, aproximadamente, 900 linhas de código fonte, ocupando 32 Kbytes de memória RAM para a sua execução.

Com relação ao módulo de geração automática de estados de ALLOS, a linguagem utilizada para o seu desenvolvimento foi *Arity/Prolog*, versão 5.0 [ARITY CORP., 1988]. O uso de *Prolog* facilitou sobremaneira o desenvolvimento deste módulo, que contém aproximadamente 350 linhas de código fonte e 84 Kbytes de código executável. Conforme discutido em [DINIZ, 1991; SILVA, 1988] são três os motivos principais desta escolha: *Prolog* não exige a especificação dos tipos de dados utilizados, permitindo inteira liberdade na descrição dos estados dos modelos; *Prolog* fornece unificação, uma forma poderosa de inicialização de variáveis e casamento de padrões usados na verificação das pré-condições; finalmente, *Prolog* possui "backtrack": quando mais de uma regra tem suas pré-condições satisfeitas ao mesmo tempo, todas elas são testadas para encontrar todos os estados alcançáveis.

O módulo solução de ALLOS foi implementado em linguagem de programação C++, versão 2.1 da ZORTECH, sendo composto de, aproximadamente, 400 linhas de código fonte, ocupando 28 Kbytes de memória para sua execução.

A opção pela linguagem C++ para o desenvolvimento do módulo responsável pela adaptação de parâmetros e do módulo solução de ALLOS deu-se pelo fato desta linguagem ser bastante eficiente na geração de código executável, compacto e rápido [ECKEL, 1989] além de facilitar, sobremaneira, o transporte para outros ambientes operacionais.

Neste sentido, a escolha das linguagens de programação C++ e *Prolog* permitiu maior flexibilidade no desenvolvimento de ALLOS, tornando o seu software modular e reutilizável, de forma a permitir, facilmente, extensões que venham a enriquecer as suas opções de utilização.

4 Validação

Neste capítulo é mostrado o processo de validação de *ALLOS* através de exemplos que englobam os elementos de modelagem disponíveis para os usuários.

Os resultados dos modelos apresentados nos exemplos que seguem, solucionados através de *ALLOS*, são comparados com resultados obtidos através do uso de técnicas analíticas, baseadas na Teoria das Filas [KLEINROCK, 1975], e através de Simulação Digital [CABRAL, 1994; CONCEIÇÃO FILHO, 1993].

4.1 Validação através de Técnicas Analíticas

Esta seção tem como finalidade comparar os resultados obtidos através de *ALLOS* com aqueles obtidos por meio de técnicas analíticas baseadas na Teoria das Filas. Este conjunto de modelos abrange os sistemas de filas associados à nomenclatura $M/M/m/K$ [KLEINROCK, 1975], sistemas de redes de filas que podem ser solucionados pelo Teorema de Burke [BURKE, 1956] (redes de filas abertas sem realimentação) e Análise do Valor Médio - AVM [REISER, 1979; REISER, 1980]. Informações detalhadas sobre essas técnicas podem ser encontradas em [KLEINROCK, 1975; MOURA, 1986].

4.1.1 Sistema $M/M/1/5$

A Figura 4.1 ilustra o modelo para o sistema $M/M/1/5$. Esse modelo é constituído por uma fonte, por uma estação de serviço, contendo um único servidor e por um sorvedouro.

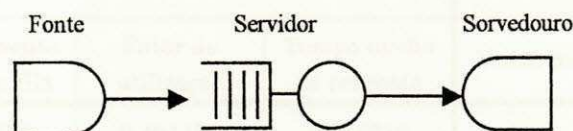


FIGURA 4.1: Modelo para o Sistema $M/M/1/5$

Para modelar este sistema, adotam-se os seguintes parâmetros para os elementos de modelagem:

Fonte	Distribuição de serviço	média
Fonte	exponencial	1.0

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Servidor	simples	5	Único	FCFS	exponencial	0.5

Sorvedouro
Sorvedouro

Classe	Prioridade
classe	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe	Fonte >> Servidor >> Sorvedouro

Após solucionar o modelo, utilizando o método apresentado na seção 3.3, *ALLOS* fornece aos usuários o vetor de probabilidades de estados para o modelo do sistema M/M/1/5, conforme apresentado na Figura 4.2.

P[0]= 0.508805	P[1]= 0.254056	P[2]= 0.126896
P[3]= 0.063102	P[4]= 0.031485	P[5]= 0.015656

FIGURA 4.2: Vetor de probabilidades para o modelo do sistema M/M/1/5

As medidas de desempenho para o modelo de um sistema M/M/1/5 são obtidas em função do vetor de probabilidades de estados, conforme apresentadas na seção 2.4. Estes resultados (Tabela 4.1) podem ser comparados com os resultados obtidos analiticamente para o sistema M/M/1/5 [KLEINROCK, 1975].

TABELA 4.1: Medidas de desempenho para o modelo do sistema M/M/1/5

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
Servidor	0.901374	0.491195	0.915710		
rotal				0.984344	0.915710

4.1.2 Sistema M/M/3/10

A Figura 4.3 representa o modelo para o sistema M/M/3/10. Esse modelo é constituído por uma fonte, por uma estação de serviço, contendo três servidores (servidor múltiplo) e por um sorvedouro, que elimina os clientes do sistema.

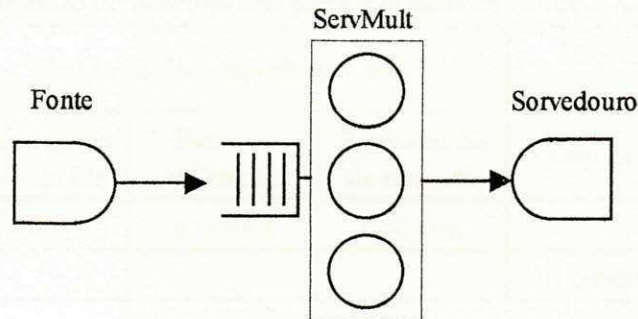


FIGURA 4.3: Modelo para o Sistema M/M/3/10

Para modelar este sistema, adotam-se os seguintes parâmetros para os elementos de modelagem:

Fonte	Distribuição de serviço	média
Fonte	exponencial	1.0

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
ServMult	múltiplo	10	3	FCFS	exponencial	0.5

Sorvedouro
Sorvedouro

Classe	Prioridade
classe	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe	Fonte >> ServMult >> Sorvedouro

O vetor de probabilidades de estados para o modelo do sistema M/M/3/10 é apresentado na Figura 4.4.

P[00]= 0.607253	P[01]= 0.302474	P[02]= 0.075313
P[03]= 0.012491	P[04]= 0.002064	P[05]= 0.000339
P[06]= 0.000055	P[07]= 0.000009	P[08]= 0.000001
P[09]= 0.000000	P[10]= 0.000000	

FIGURA 4.4: Vetor de probabilidades para o modelo do sistema M/M/3/10

Seguem as medidas de desempenho para o modelo do sistema M/M/3/10. Estes resultados (Tabela 4.2) podem ser comparados com os resultados obtidos analiticamente para o sistema M/M/3/10 [KLEINROCK, 1975].

TABELA 4.2: Medidas de desempenho para o modelo do sistema M/M/3/10

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
ServMult	0.500933	0.165994	0.500933		
rotal				1.000000	0.500933

4.1.3 Modelo de uma Rede Aberta sem Realimentação

O modelo apresentado na Figura 4.5, objetiva validar a ferramenta ALLOS para modelos de redes abertas sem realimentação. Esse modelo é constituído por uma fonte, por duas estações de serviço conectadas em série (a primeira contém apenas um servidor e a segunda dois servidores) e por um sorvedouro.

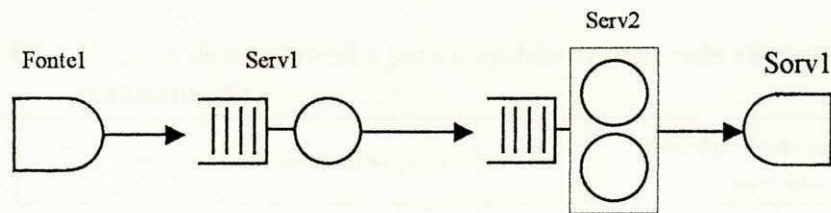


FIGURA 4.5: Modelo de uma rede aberta sem realimentação

Para modelar esta rede definem-se os seguintes parâmetros para os elementos de modelagem:

Fonte	Distribuição de serviço	média
Fonte1	exponencial	0.5

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Serv1	simplex	4	Único	FCFS	exponencial	0.5
Serv2	múltiplo	3	2	FCFS	exponencial	1.0

Sorvedouro
Sorv1

Classe	Prioridade
classe1	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe1	Fonte1 >> Serv1 >> Serv2 >> Sorv1

O vetor de probabilidades de estados para o modelo de uma rede aberta sem realimentação é mostrado na Figura 4.6.

P[0,0]= 0.015665	P[1,0]= 0.017275	P[2,0]= 0.021761	P[0,1]= 0.031008
P[1,1]= 0.037470	P[3,0]= 0.032005	P[2,1]= 0.052267	P[0,2]= 0.028998
P[4,0]= 0.083122	P[1,2]= 0.040703	P[3,1]= 0.084410	P[2,2]= 0.061093
P[0,3]= 0.020380	P[4,1]= 0.102441	P[1,3]= 0.040747	P[3,2]= 0.075708
P[2,3]= 0.058187	P[4,2]= 0.069495	P[3,3]= 0.063752	P[4,3]= 0.063513

FIGURA 4.6: Vetor de probabilidades para o modelo de uma rede aberta sem realimentação

Estes resultados (Tabela 4.3) podem ser comparados com os resultados obtidos analiticamente através do Teorema de Burke [BURKE, 1956].

Seguem as medidas de desempenho obtidas para o modelo de uma rede aberta sem realimentação:

TABELA 4.3: Medidas de desempenho para o modelo de uma rede aberta sem realimentação

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
Serv1	2.564721	0.903949	1.881869		
Serv2	1.599327	0.676374	1.173509		
rotal				1.362858	3.055378

4.1.4 Modelo de uma Rede Fechada

Para validar ALLOS, solucionando modelos de redes de filas fechadas, escolheu-se um modelo contendo duas estações de serviço conectadas em série, onde cada estação possui um único servidor. No modelo existe um número fixo de clientes (população) circulando. Esse modelo é mostrado na Figura 4.7.

Para modelar esta rede definem-se os seguintes parâmetros para os elementos de modelagem:

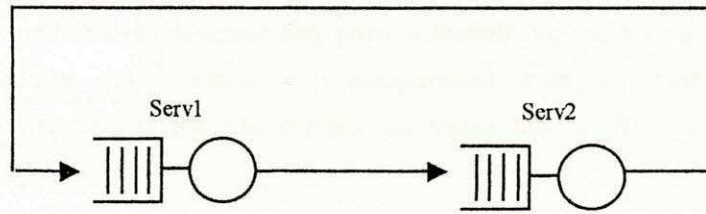


FIGURA 4.7: Modelo de uma rede fechada

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Serv1	simples	ilimitado	Único	FCFS	exponencial	0.5
Serv2	simples	ilimitado	Único	FCFS	exponencial	1.0

Classe	Prioridade
classe1	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe1	Serv1 >> Serv2 >> Serv1

Por ser uma rota fechada, a interface solicita a população da rede, tendo sido atribuído o valor 5. O vetor de probabilidades associados ao modelo de uma rede fechada pode ser visto na Figura 4.8.

Pode-se observar, ainda, que o comprimento das filas dos servidores foram especificados como ilimitado. Esta especificação não constitui um erro, pois para modelos de redes fechadas, *ALLOS* atribui o valor da população ao limite das filas do modelo.

$P[5,0]= 0.167552$	$P[4,1]= 0.167401$	$P[3,2]= 0.166817$
$P[2,3]= 0.166517$	$P[1,4]= 0.165932$	$P[0,5]= 0.165782$

FIGURA 4.8: Vetor de probabilidades para o modelo de uma rede fechada

TABELA 4.4: Medidas de desempenho para o modelo de uma rede fechada

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Tempo de circulação
Serv1	2.506779	0.834218	1.502472		
Serv2	2.493221	0.832448	1.494346		
rota1				1.668436	2.996818

Seguem-se as medidas de desempenho para o modelo de uma rede fechada. Estes resultados (Tabela 4.4) podem ser comparados com os resultados obtidos analiticamente através do método Análise do Valor Médio [REISER, 1979; REISER, 1980].

4.1.5 Modelo de uma Rede Aberta com Ponto de Multiplicação

O modelo de uma rede aberta que segue foi selecionado com o propósito de validar o elemento de modelagem *ponto de multiplicação*, conforme pode ser visto na Figura 4.9. Para esse modelo tem-se três estações de serviço, cada uma contendo um único servidor, um ponto de multiplicação e dois sorvedouros.

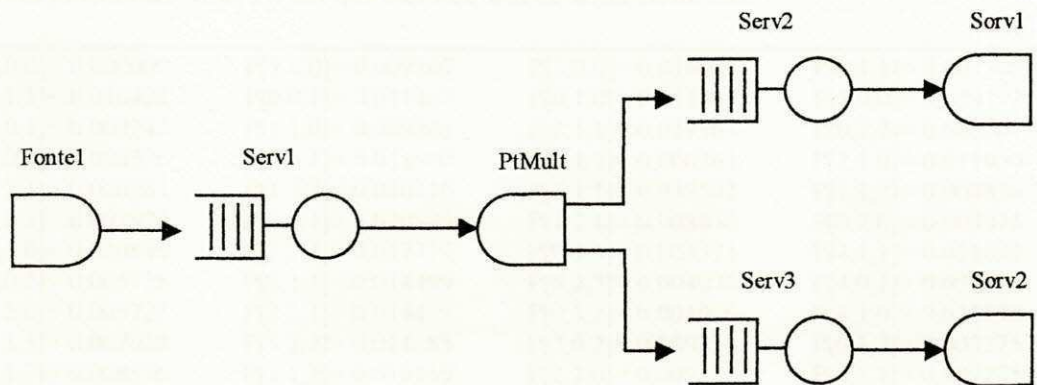


FIGURA 4.9: Rede aberta com ponto de multiplicação

Os parâmetros adotados na definição dos elementos de modelagem para o modelo de uma rede aberta com ponto de multiplicação são os seguintes:

Fonte	Distribuição de serviço	média
Fonte1	exponencial	0.5

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Serv1	simples	4	Único	FCFS	exponencial	0.5
Serv2	simples	3	Único	FCFS	exponencial	0.5
Serv3	simples	3	Único	FCFS	exponencial	0.5

Sorvedouro
Sorv1
Sorv2

Ponto de multiplicação	Nº de saídas
PtMult	2

Classe	Prioridade
classe1	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe1	Fonte1 >> Serv1 >> PtMult >> Serv2 >> Sorv1
Rota2	classe1	Fonte1 >> Serv1 >> PtMult >> Serv3 >> Sorv2

A Figura 4.10 apresenta o vetor de probabilidades de estados associados ao modelo de uma rede aberta com ponto de multiplicação.

De forma similar ao modelo de redes de filas abertas sem realimentação, pode-se verificar que o processo de saída da estação de serviço Serv1 é um processo de Poisson com parâmetro λ , de forma que Serv1 é um sistema M/M/1/K.

P[0,0,0]= 0.000000	P[1,0,0]= 0.009307	P[2,0,0]= 0.016624	P[0,1,1]= 0.007309
P[1,1,1]= 0.010422	P[0,0,1]= 0.011467	P[0,1,0]= 0.011467	P[3,0,0]= 0.029193
P[1,0,1]= 0.009243	P[1,1,0]= 0.009243	P[2,1,1]= 0.017163	P[0,2,2]= 0.006207
P[4,0,0]= 0.094576	P[2,0,1]= 0.011900	P[0,1,2]= 0.006263	P[2,1,0]= 0.011900
P[0,2,1]= 0.006263	P[1,2,2]= 0.010116	P[3,1,1]= 0.037502	P[1,1,2]= 0.008836
P[0,0,2]= 0.003975	P[3,0,1]= 0.020849	P[1,2,1]= 0.008836	P[0,2,0]= 0.003975
P[3,1,0]= 0.020849	P[2,2,2]= 0.017717	P[0,3,3]= 0.003375	P[4,1,1]= 0.021092
P[1,0,2]= 0.005728	P[2,1,2]= 0.014499	P[0,2,3]= 0.004076	P[4,0,1]= 0.032790
P[1,2,0]= 0.005728	P[2,2,1]= 0.014499	P[0,3,2]= 0.004076	P[4,1,0]= 0.032790
P[1,3,3]= 0.007028	P[3,2,2]= 0.014685	P[2,0,2]= 0.009236	P[0,1,3]= 0.003275
P[1,2,3]= 0.008536	P[3,1,2]= 0.019169	P[2,2,0]= 0.009236	P[0,3,1]= 0.003275
P[1,3,2]= 0.008536	P[3,2,1]= 0.019169	P[2,3,3]= 0.007232	P[4,2,2]= 0.009109
P[1,1,3]= 0.007020	P[0,0,3]= 0.001643	P[3,0,2]= 0.013159	P[2,2,3]= 0.011637
P[4,1,2]= 0.012918	P[1,3,1]= 0.007020	P[0,3,0]= 0.001643	P[3,2,0]= 0.013159
P[2,3,2]= 0.011637	P[4,2,1]= 0.012918	P[3,3,3]= 0.005441	P[1,0,3]= 0.004339
P[2,1,3]= 0.010600	P[4,0,2]= 0.023743	P[3,2,3]= 0.009987	P[1,3,0]= 0.004339
P[2,3,1]= 0.010600	P[4,2,0]= 0.023743	P[3,3,2]= 0.009987	P[4,3,3]= 0.002715
P[2,0,3]= 0.007469	P[3,1,3]= 0.014758	P[4,2,3]= 0.006338	P[2,3,0]= 0.007469
P[3,3,1]= 0.014758	P[4,3,2]= 0.006338	P[3,0,3]= 0.011098	P[4,1,3]= 0.010525
P[3,3,0]= 0.011098	P[4,3,1]= 0.010525	P[4,0,3]= 0.021516	P[4,3,0]= 0.021516

FIGURA 4.10: Vetor de probabilidades para o modelo de uma rede aberta com ponto de multiplicação

Pode-se perceber que o fluxo de clientes entre Serv1 e Serv2 não têm suas características alteradas pelo ponto de multiplicação, desta forma, pelo Teorema de Burke pode-se concluir que a estação de serviço Serv2 é também um sistema M/M/1/K. Por analogia, conclui-se que a estação de serviço Serv3 também é um sistema M/M/1/K.

As medidas de desempenho para o modelo de uma rede aberta com ponto de multiplicação podem ser vistas, abaixo. No processo de validação desse modelo, os resultados de Serv1, Serv2 e Serv3 (Tabela 4.5) são comparados com os resultados obtidos analiticamente para o sistema M/M/1/K [KLEINROCK, 1975].

TABELA 4.5: Medidas de desempenho para o modelo de uma rede aberta com ponto de multiplicação

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
Serv1	2.670307	0.921710	2.032667		
Serv2	1.195299	0.662145	0.909874		
Serv3	1.195299	0.662145	0.909874		
rota1				1.313696	2.942541
rota2				1.313696	2.942541

4.1.6 Modelo de uma Rede Aberta com Ponto de Fusão

Para ilustrar o uso do elemento de modelagem *ponto de fusão* escolheu-se um modelo de uma rede com duas fontes, três estações de serviço, onde duas são do tipo servidor único e a outra, servidor múltiplo com dois servidores, um ponto de fusão e um sorvedouro. Esse modelo pode ser visto na Figura 4.11.

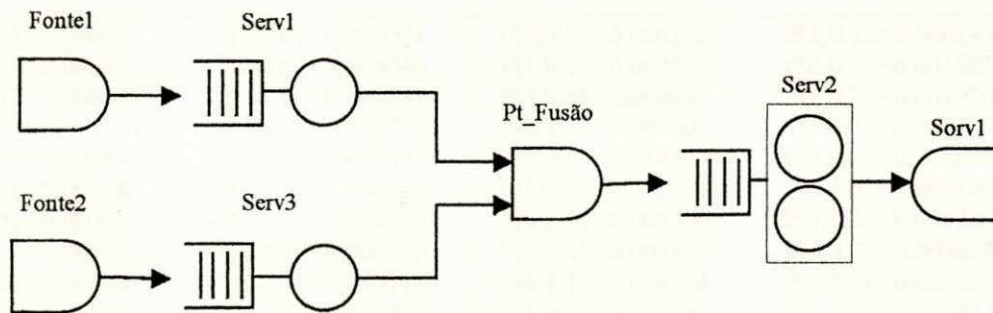


FIGURA 4.11: Modelo de uma rede aberta com ponto de fusão

Os parâmetros adotados na definição dos elementos de modelagem para o modelo de uma rede aberta com ponto de fusão são os seguintes:

Fonte	Distribuição de serviço	média
Fonte1	exponencial	0.5
Fonte2	exponencial	0.25

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Serv1	simples	4	Único	FCFS	exponencial	0.5
Serv2	múltiplo	3	2	FCFS	exponencial	1.0
Serv3	simples	4	Único	FCFS	exponencial	0.25

Servidouro	
Sorv1	

Ponto de fusão	Nº de entradas
Pt_Fusao	2

Classe	Prioridade
classe1	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe1	Fonte1 >> Serv1 >> Pt_Fusão >> Serv2 >> Sorv1
Rota2	classe1	Fonte2 >> Serv3 >> Pt_Fusão

Nesse modelo é necessário que um cliente tenha concluído serviço em Serv1 e outro tenha concluído serviço em Serv3 para que o ponto de fusão proceda uma fusão.

Pode-se observar que os parâmetros adotados em Fonte2 e Serv3 fazem com que Pt_Fusão sempre tenha um cliente em Serv3 esperando a chegada de um cliente em Serv1. Dessa forma o fluxo de saída de clientes de Pt_Fusão é determinado pelo fluxo de saída de Serv1. Nota-se ainda, que a rota "rota1" corresponde a uma cadeia aberta sem realimentação (compare com Figura 4.5), podendo-se aplicar o Teorema de Burke.

P[0,0,0]= 0.000000	P[1,0,0]= 0.000034	P[0,0,1]= 0.000102	P[1,0,1]= 0.000143
P[0,0,2]= 0.000425	P[2,0,0]= 0.000045	P[1,0,2]= 0.000527	P[0,0,3]= 0.001685
P[2,0,1]= 0.000157	P[0,1,0]= 0.000153	P[3,0,0]= 0.000042	P[1,0,3]= 0.001970
P[0,0,4]= 0.013764	P[2,0,2]= 0.000725	P[0,1,1]= 0.000609	P[1,1,0]= 0.000206
P[3,0,1]= 0.000137	P[4,0,0]= 0.000047	P[1,0,4]= 0.014974	P[2,0,3]= 0.002689
P[0,1,2]= 0.002140	P[1,1,1]= 0.000803	P[3,0,2]= 0.000671	P[2,1,0]= 0.000199
P[4,0,1]= 0.000136	P[2,0,4]= 0.018505	P[0,1,3]= 0.008360	P[1,1,2]= 0.002786
P[3,0,3]= 0.004562	P[2,1,1]= 0.000789	P[0,2,0]= 0.000318	P[4,0,2]= 0.000615
P[3,1,0]= 0.000161	P[1,1,3]= 0.010246	P[0,1,4]= 0.020284	P[3,0,4]= 0.026827
P[2,1,2]= 0.004120	P[0,2,1]= 0.001295	P[4,0,3]= 0.003944	P[1,2,0]= 0.000411
P[3,1,1]= 0.000615	P[4,1,0]= 0.000104	P[1,1,4]= 0.024011	P[2,1,3]= 0.014654
P[0,2,2]= 0.004286	P[4,0,4]= 0.078377	P[1,2,1]= 0.001869	P[3,1,2]= 0.003377
P[2,2,0]= 0.000355	P[4,1,1]= 0.000357	P[2,1,4]= 0.032936	P[0,2,3]= 0.009894
P[1,2,2]= 0.006093	P[3,1,3]= 0.028442	P[2,2,1]= 0.001681	P[0,3,0]= 0.000467
P[4,1,2]= 0.001809	P[3,2,0]= 0.000229	P[1,2,3]= 0.013581	P[0,2,4]= 0.013419
P[3,1,4]= 0.051877	P[2,2,2]= 0.009621	P[0,3,1]= 0.001758	P[4,1,3]= 0.012100
P[1,3,0]= 0.000537	P[3,2,1]= 0.001044	P[4,2,0]= 0.000099	P[1,2,4]= 0.018998
P[2,2,3]= 0.020594	P[0,3,2]= 0.004278	P[4,1,4]= 0.087663	P[1,3,1]= 0.003112
P[3,2,2]= 0.005912	P[2,3,0]= 0.000395	P[4,2,1]= 0.000428	P[2,2,4]= 0.028873
P[0,3,3]= 0.006898	P[1,3,2]= 0.007773	P[3,2,3]= 0.028268	P[2,3,1]= 0.002452
P[4,2,2]= 0.002248	P[3,3,0]= 0.000205	P[1,3,3]= 0.012832	P[0,3,4]= 0.006919
P[3,2,4]= 0.039926	P[2,3,2]= 0.010227	P[4,2,3]= 0.010328	P[3,3,1]= 0.001276
P[4,3,0]= 0.000068	P[1,3,4]= 0.016310	P[2,3,3]= 0.018290	P[4,2,4]= 0.055796
P[3,3,2]= 0.005771	P[4,3,1]= 0.000470	P[2,3,4]= 0.026424	P[3,3,3]= 0.021377
P[4,3,2]= 0.002234	P[3,3,4]= 0.034509	P[4,3,3]= 0.008598	P[4,3,4]= 0.051351

FIGURA 4.12: Vetor de probabilidades para o modelo de uma rede aberta com ponto de fusão

A Figura 4.12 apresenta o vetor de probabilidades de estados associados ao modelo de uma rede aberta com ponto de fusão.

Seguem as medidas de desempenho para o modelo de uma rede aberta com ponto de fusão. Estes resultados (Tabela 4.6) podem ser comparados com resultados obtidos analiticamente para o sistema M/M/1/K.

TABELA 4.6: Medidas de desempenho para o modelo de uma rede aberta com ponto de fusão

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
Serv1	2.557450	0.902946	1.871593		
Serv2	1.593526	0.674498	1.166174		
Serv3	3.535418	0.995925	2.612967		
rota1				1.366456	3.047767
rota2				1.353028	2.612967

4.1.7 Modelo do Protocolo de Sessão com Diálogo Semi-Duplex

Na Figura 4.13 considera-se um modelo de uma conexão de sessão com diálogo semi-duplex [CABRAL, 1987]. Nesse modelo, a taxa média de chegada é de 12 Unidades de Dados do Serviço de Sessão (UDSSs) por segundo, na entidade de sessão de referência. A capacidade de armazenamento desta entidade é de 6 Unidades de Dados do Protocolo de Sessão (UDPSs). Adota-se a disciplina de liberação de UDPSs do tipo limitada, permitindo liberar até 2 UDPSs. A taxa média de transmissão na conexão de transporte associada a essa conexão de sessão é de 10 Unidades de Dados do Serviço de Transporte (UDSTs) por segundo. A conexão de transporte tem capacidade de armazenamento igual a 6 Unidades de Dados do Protocolo de Transporte (UDPTs).

A entidade de sessão de referência somente envia uma UDPS à entidade de sessão par quando recebe a ficha de dados. Na entidade de sessão par tem-se uma taxa média de chegada de 20 UDSSs por segundo. A distribuição de probabilidade considerada é exponencial.

A entidade de sessão de referência transferindo UDPSs libera a ficha de dados juntamente com a última UDPSs emitida. A entidade de sessão par, à medida que recebe UDPSs entrega-as (como UDSSs) ao usuário SS receptor. A entrega da última UDSS é seguida da geração de uma UDSS de resposta (e/ou reconhecimento). Essa

UDSS, juntamente com a ficha de dados, são enviadas pela conexão de sessão à entidade de sessão de referência.

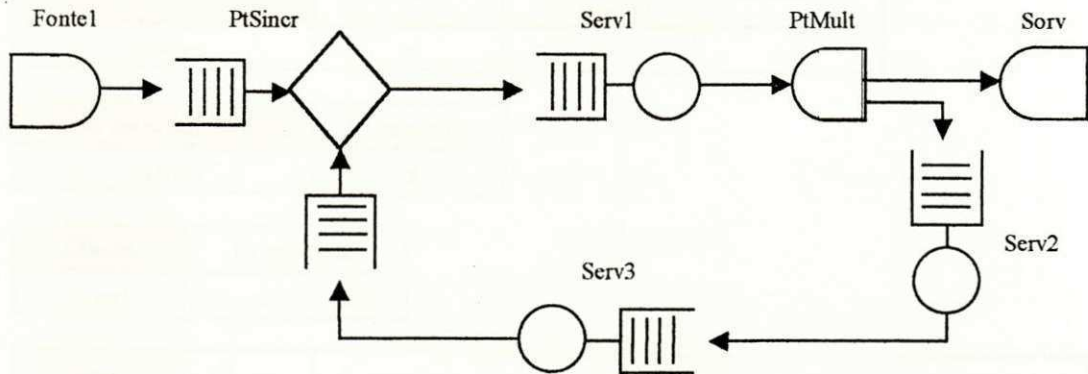


FIGURA 4.13: Modelo do protocolo de sessão com diálogo semi-duplex

A entidade de sessão de referência é modelada pelos seguintes elementos:

- uma fonte (Fonte1) que representa a geração de UDSSs;
- um ponto de sincronização (PtSincr) que representa o processo de liberação de UDPSs com a ficha de dados para a entidade de sessão par.

A entidade de sessão par é modelada pelos seguintes elementos:

- um ponto de multiplicação (PtMult)
- um sorvedouro (Sorv)
- uma estação de serviço (Serv2)

Finalmente, a conexão de transporte, associada à conexão de sessão, é modelada por duas estações de serviço (Serv1 e Serv3).

Os parâmetros adotados na definição dos elementos de modelagem para o modelo do protocolo de sessão com diálogo semi-duplex são os seguintes:

Fonte	Distribuição de serviço	média
Fonte1	exponencial	0.08333

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Serv1	simples	6	Único	FCFS	exponencial	0.1
Serv2	simples	6	Único	FCFS	exponencial	0.05
Serv3	simples	6	Único	FCFS	exponencial	0.1

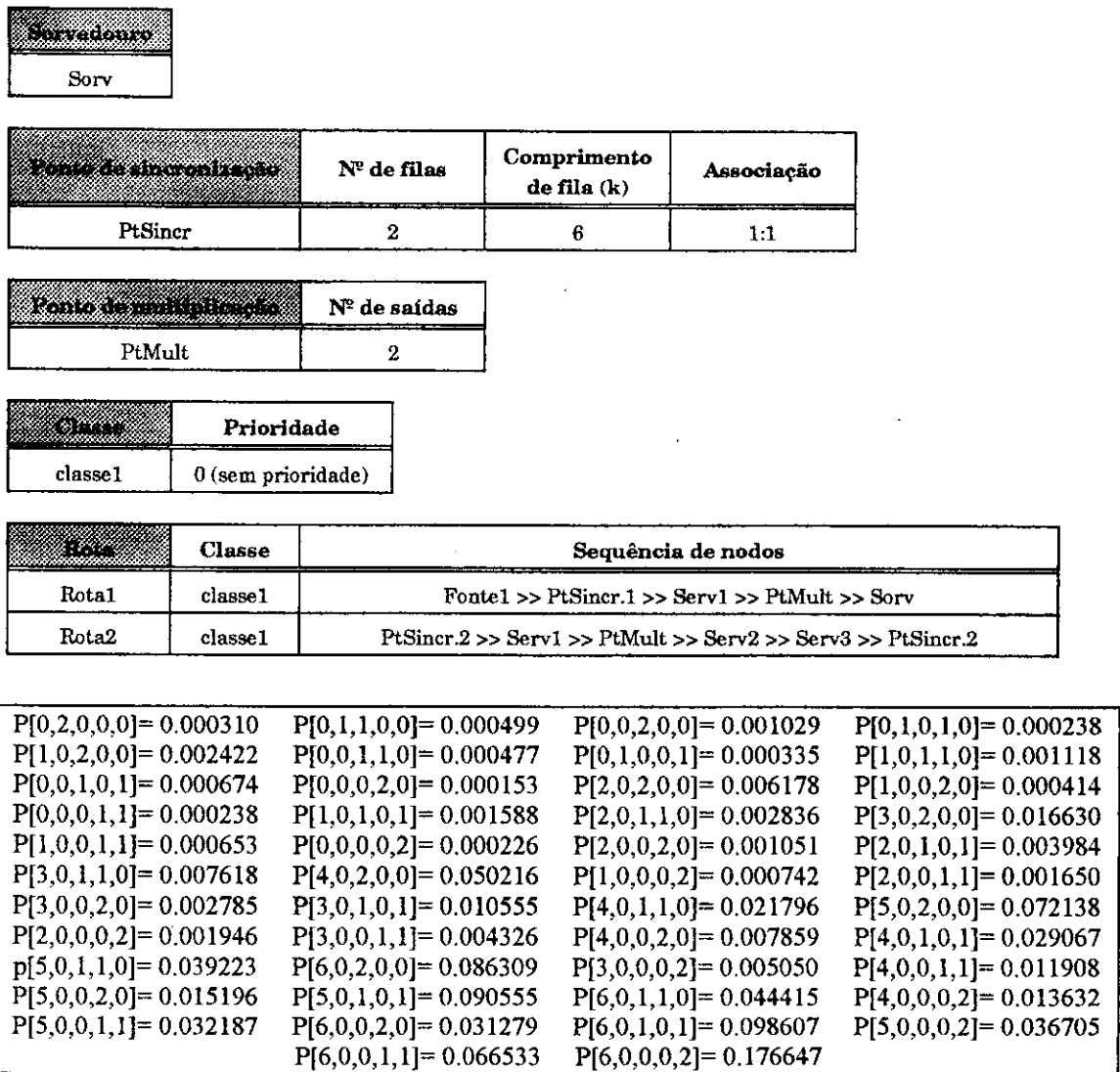


FIGURA 4.14: Vetor de probabilidades para o modelo do protocolo de sessão com diálogo semi-duplex

TABELA 4.7: Medidas de desempenho para o modelo do protocolo de sessão com diálogo semi-duplex

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
PtSincr.1	5.173801		0.868167		
PtSincr.2	0.001691		0.000283		
Serv1	0.822857	0.587934	0.138075		
Serv2	0.352692	0.293955	0.059181		
Serv3	0.822758	0.587810	0.138059		
rota1				6.959452	1.006242

A Figura 4.14 mostra o vetor de probabilidades de estados associados ao modelo do protocolo de sessão com diálogo semi-duplex.

As medidas de desempenho (Tabela 4.7) para o modelo do protocolo de sessão com diálogo semi-duplex foram comparadas com os resultados apresentados em [Cabral, 1987].

4.1.8 Modelo do Protocolo de Sessão com Quarentena de Dados e Modo de Diálogo Duplex

Na Figura 4.15, considera-se um modelo do protocolo de sessão com quarentena de dados local e modo de diálogo duplex [CABRAL, 1987]. Nesse modelo, UDSSs chegam à entidade de sessão de referência e são quarentenados (como UDPSs) até a composição de uma Unidade de Quarentena de Dados (UQD), quando então podem ser liberadas e enviadas pela conexão de sessão à entidade de sessão par. UDPSs que chegam à entidade de sessão par, são entregues ao usuário SS receptor. Uma UDPS corresponde a uma UDSS.

A capacidade de armazenamento na entidade de sessão de referência é de 6 UDPSs. A taxa média de transmissão na conexão de transporte atribuída a esta conexão de sessão é de 10 UDSTs por segundo. A conexão de transporte tem capacidade de armazenamento de 6 UDSTs.

Nessa Figura, a fonte (Fonte) representa a geração de UDSSs, o ponto de sincronização (Pt_Sincr) representa o processo de formação e liberação de uma UQD, a estação de serviço (Serv) representa a conexão de transporte e o sorvedouro (Sorv) representa a entrega de UDSSs à entidade de sessão par.

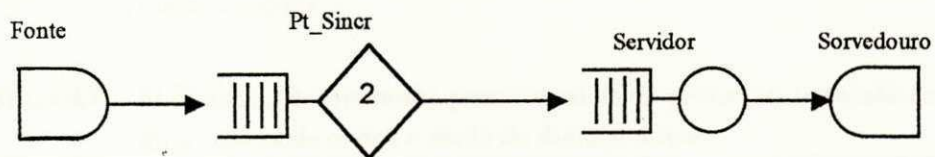


FIGURA 4.15: Modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex

Os parâmetros adotados na definição dos elementos de modelagem para o modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex são os seguintes:

Fonte	Distribuição de serviço	média
Fonte	exponencial	0.08333

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Servidor	simples	6	Único	FCFS	exponencial	0.1

Sorvedouro
Sorvedouro

Ponto de sincronização	Nº de filas	Comprimento de fila (k)	Associação
Pt_Sincr	1	6	2:2

Classe	Prioridade
classe1	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe1	Fonte >> Pt_Sincr.1 >> Servidor >> Sorvedouro

O vetor de probabilidades de estados associado ao modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex pode ser visto de acordo com a Figura 4.16.

P[0,0]= 0.007797	P[1,0]= 0.020666	P[0,2]= 0.020534	P[1,2]= 0.022574
P[0,1]= 0.009335	P[1,1]= 0.015356	P[0,4]= 0.026103	P[0,3]= 0.020285
P[1,4]= 0.030610	P[1,3]= 0.025040	P[0,6]= 0.036833	P[0,5]= 0.030367
P[1,6]= 0.042625	P[1,5]= 0.035901	P[2,6]= 0.053936	P[2,5]= 0.044144
P[3,6]= 0.056245	P[3,5]= 0.049699	P[4,6]= 0.089320	P[4,5]= 0.067633
P[5,6]= 0.048673	P[5,5]= 0.058963	P[6,6]= 0.058342	P[6,5]= 0.129022

FIGURA 4.16: Modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex

TABELA 4.8: Medidas de desempenho para o modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex

Elemento de modelagem	Medidas de desempenho por nodo			Medidas de desempenho por rota	
	Comprimento médio de fila	Fator de utilização	Tempo médio de resposta	Vazão média	Atraso médio fim-a-fim
Pt_Sincr.1	2.996937		0.305416		
Servidor (Rede)	4.868222	0.971540	0.496117		
rotal				9.751632	0.801533

As medidas de desempenho apresentadas a seguir (Tabela 4.8), para o modelo do protocolo de sessão com quarentena de dados e modo de diálogo duplex, foram comparadas com os resultados apresentados em [Cabral, 1987].

4.2 Validação através de Simulação Digital

Nesta seção, o modelo validado apresenta um grau de complexidade maior que aqueles apresentados na seção anterior, dificultando a validação deste por meio de técnicas analíticas. Desta forma, sua validação foi feita mediante a comparação dos seus resultados com aqueles obtidos através da técnica de simulação digital [CONCEIÇÃO FILHO, 1993].

4.2.1 Modelo de uma Rede Local Anel com Passagem de Ficha

Na Figura 4.17 tem-se o modelo de uma rede local anel com passagem de ficha, com três interfaces. O número restrito de interfaces deve-se apenas à necessidade de minimizar o espaço de estados para o modelo apresentado. Nesse modelo, Fonte_1, $i = 1, 2, 3$ representam as fontes de pacotes de dados. As interfaces, juntamente com o protocolo de acesso (com passagem de ficha) são modelados por um ponto escalonador, PtEsc, com disciplina de liberação cíclica. Pacotes chegam em cada fila do ponto escalonador através das fontes Fonte_1 pertencentes a classe de dados *classe1*.

A sub-rede de comunicação é representada por uma estação de trabalho (servidor simples), denominada Rede. Pacotes servidos pelo nodo Rede entram no nodo Sorvedouro, onde são eliminados do modelo. Isto corresponde à transmissão desses pacotes pela sub-rede de comunicação e suas distribuições para as próprias interfaces de destino.

Este exemplo ilustra o uso do elemento de modelagem *ponto escalonador*.

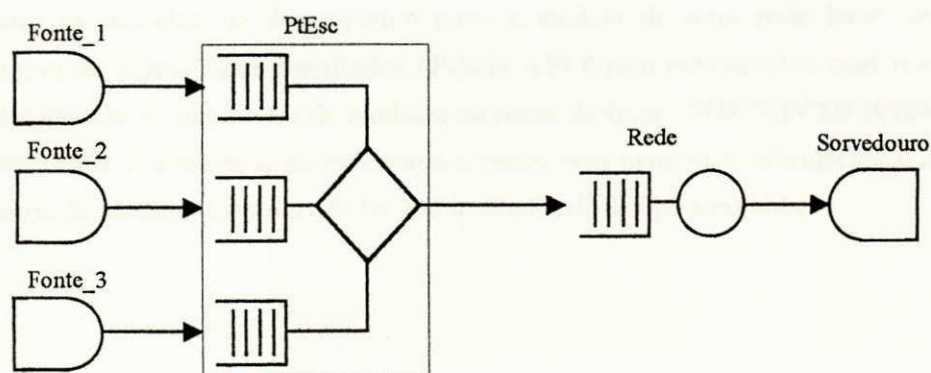


FIGURA 4.17: Modelo de uma rede local anel com passagem de ficha

Os parâmetros adotados na definição dos elementos de modelagem para o modelo de uma rede local anel com passagem de ficha são os seguintes:

Clientes
Pacotes de dados

Fonte	Distribuição de serviço	média
Fonte_1	exponencial	0.025
Fonte_2	exponencial	0.025
Fonte_3	exponencial	0.025

Estação de serviço	Tipo	Comprimento de fila (k)	Quantidade	Disciplina de fila	Distribuição de serviço	Média
Rede	simples	1	Único	FCFS	exponencial	0.01

Sorvedouro
Sorvedouro

Ponto escalonador	Nº de filas	Disciplina de liberação	Comprimento de fila (k)
PtEsc	1	cíclica (1,2,3)	3

Classe	Prioridade
classe1	0 (sem prioridade)

Rota	Classe	Sequência de nodos
Rota1	classe1	Fonte_1 >> PtEsc.1 >> Rede >> Sorvedouro
Rota2	classe1	Fonte_2 >> PtEsc.2 >> Rede >> Sorvedouro
Rota3	classe1	Fonte_3 >> PtEsc.3 >> Rede >> Sorvedouro

A Figura 4.18 apresenta o vetor de probabilidades de estados associados ao modelo de uma rede local anel com passagem de ficha.

Seguem as medidas de desempenho para o modelo de uma rede local anel com passagem de ficha. Estes resultados (Tabela 4.9) foram comparados com resultados obtidos através do simulador de modelos de redes de filas - *SIM/SAVAD* [CONCEIÇÃO FILHO, 1993]. A simulação foi executada 5 vezes, com nível de confiança igual a 90% e o critério de término considerado foi 400 unidades de tempo simulado.

P[0,0,0,0]= 0.059455	P[0,1,0,0]= 0.071118	P[1,1,0,0]= 0.028284	P[0,1,1,0]= 0.028284
P[0,1,0,1]= 0.028284	P[1,1,0,1]= 0.018671	P[0,1,1,1]= 0.018671	P[0,1,0,2]= 0.015036
P[1,1,1,0]= 0.018671	P[0,1,2,0]= 0.015036	P[2,1,0,0]= 0.015036	P[1,1,0,2]= 0.012947
P[0,1,1,2]= 0.012947	P[0,1,0,3]= 0.008794	P[1,1,1,1]= 0.016371	P[0,1,2,1]= 0.012947
P[2,1,0,1]= 0.012947	P[1,1,2,0]= 0.012947	P[0,1,3,0]= 0.008794	P[2,1,1,0]= 0.012947
P[3,1,0,0]= 0.008794	P[1,1,0,3]= 0.009813	P[0,1,1,3]= 0.009813	P[1,1,1,2]= 0.013592
P[0,1,2,2]= 0.011113	P[2,1,0,2]= 0.011113	P[1,1,2,1]= 0.013592	P[0,1,3,1]= 0.009813
P[2,1,1,1]= 0.013592	P[3,1,0,1]= 0.009813	P[1,1,3,0]= 0.009813	P[2,1,2,0]= 0.011113
P[3,1,1,0]= 0.009813	P[1,1,1,3]= 0.012405	P[0,1,2,3]= 0.009666	P[2,1,0,3]= 0.009666
P[1,1,2,2]= 0.013284	P[0,1,3,2]= 0.009666	P[2,1,1,2]= 0.013284	P[3,1,0,2]= 0.009666
P[1,1,3,1]= 0.012405	P[2,1,2,1]= 0.013284	P[3,1,1,1]= 0.012405	P[2,1,3,0]= 0.009666
P[3,1,2,0]= 0.009666	P[1,1,2,3]= 0.013582	P[0,1,3,3]= 0.009030	P[2,1,1,3]= 0.013582
P[3,1,0,3]= 0.009030	P[1,1,3,2]= 0.013582	P[2,1,2,2]= 0.014601	P[3,1,1,2]= 0.013582
P[2,1,3,1]= 0.013582	P[3,1,2,1]= 0.013582	P[3,1,3,0]= 0.009030	P[1,1,3,3]= 0.014784
P[2,1,2,3]= 0.016211	P[3,1,1,3]= 0.014784	P[2,1,3,2]= 0.016211	P[3,1,2,2]= 0.016211
P[3,1,3,1]= 0.014784	P[2,1,3,3]= 0.018783	P[3,1,2,3]= 0.018783	P[3,1,3,2]= 0.018783
P[3,1,3,3]= 0.022445			

FIGURA 4.18: Vetor de probabilidades para o modelo de uma rede local anel com passagem de ficha

TABELA 4.9: Medidas de desempenho para o modelo de uma rede local anel com passagem de ficha

Elemento de modelagem	Medidas de desempenho por nodo					
	Tempo médio de resposta (ALLOS)	Tempo médio de resposta (Simulador)	Intervalo de confiança	Fator de utilização (ALLOS)	Fator de utilização (Simulador)	Intervalo de confiança
PtEsc.1	0.041185	0.044452	[0.043596; 0.045307]			
PtEsc.2	0.041185	0.044452	[0.043596; 0.045307]			
PtEsc.3	0.041185	0.044452	[0.043596; 0.045307]			
Servidor (Rede)	0.029808	0.020038	[0.019922; 0.020155]	0.940548	0.947188	[0.942033; 0.952343]

Os resultados mostraram que ALLOS se adequa para solucionar uma grande variedade de sistemas que podem ser modelados usando o paradigma de redes de filas [OLIVEIRA, 1995a; OLIVEIRA 1995b].

ALLOS foi validada através da execução de testes cuidadosos com o propósito de comparar seus resultados com outras ferramentas já existentes, na literatura especializada, para solucionar modelos de redes de filas.

Apesar de se utilizar um conjunto finito de testes de validação, os resultados obtidos acenam para a corretude da ferramenta ALLOS, tornando-a disponível para a avaliação de desempenho de sistemas que exibem contenção de recursos.

5 Conclusões e sugestões

ALLOS é uma ferramenta para especificar e solucionar modelos de redes de filas usando cadeias de Markov. Esta ferramenta é parte integrante do módulo solução de um ambiente de simulação inteligente, denominado *SAVAD*.

Do ponto de vista do usuário, é desejável que, em uma ferramenta de modelagem, tanto a definição matemática do sistema modelado quanto os detalhes referentes às técnicas de solução sejam transparentes, de forma que o usuário não precise ser um especialista nessas técnicas. Neste sentido, a ferramenta *ALLOS*, com uma interface amigável, pode ser facilmente utilizada, não exigindo que seus usuários necessitem de conhecimentos mais aprofundados em avaliação de desempenho de sistemas, no que diz respeito à solução de modelos de redes de filas através do uso de cadeias de Markov.

ALLOS gera, automaticamente, os estados de modelos markovianos a partir de um estado inicial e das regras que descrevem o comportamento do sistema, originando a partir do espaço de estados, a matriz de transição de estados que, posteriormente, é solucionada com o objetivo de se obter as medidas de desempenho mais relevantes para um modelo markoviano, além do vetor de probabilidades de estados.

A metodologia utilizada por *ALLOS* para a obtenção do espaço de estados de um modelo é a exploração dinâmica do espaço de estados [SILVA, 1992b]. A experiência mostrou que esta metodologia é simples, porém bastante eficiente.

Em [DINIZ, 1990] foi proposta uma ferramenta que implementa o paradigma de modelagem orientado a objeto, permitindo a descrição de modelos em alto nível. Seu domínio de aplicação abrange análise de desempenho e confiabilidade de sistemas. No entanto, esta ferramenta, apesar de gerar a matriz de taxas de transição, não obtém o vetor de probabilidades associado aos estados do modelo e conseqüentemente não apresenta as medidas de desempenho. Quanto a *ALLOS*, esta restringe seu domínio de aplicação para a modelagem e avaliação de desempenho de sistemas, supondo-se que estes são perfeitamente confiáveis. Entretanto, *ALLOS* possui certas vantagens,

uma vez que soluciona os modelos markovianos e apresenta as medidas de desempenho mais relevantes.

Outra vantagem oferecida por *ALLOS* diz respeito à possibilidade de múltiplas entradas/saídas associadas aos seus elementos do tipo *ponto de controle*, que facilitam a modelagem de sistemas de redes de filas que apresentam múltiplas rotas.

ALLOS foi validada através da execução de testes cuidadosos para verificar sua coerência com outras ferramentas, já existentes na literatura especializada, para solucionar modelos de redes de filas.

A interface de *ALLOS* é bastante simples e amigável, oferecendo ao usuário facilidades para descrição de uma grande variedade de modelos que apresentam contenção de recursos; ela é útil tanto na descrição de modelos e na sua validação, quanto na apresentação das medidas de desempenho e do vetor de probabilidades, obtidos após a solução de um modelo em análise.

A escolha das linguagens *Prolog* e *C++* permitiu maior flexibilidade no desenvolvimento desta ferramenta, tornando o seu software modular e reutilizável, de forma a permitir facilmente extensões que venham a enriquecer as suas opções de utilização.

ALLOS, como ferramenta integrante do módulo solução do *SAVAD*, foi projetada para funcionamento em equipamentos de pequeno porte, como microcomputadores compatíveis com a linha IBM-PC, tornando possível a sua utilização em larga escala. Contudo, seus módulos (Adaptação, Geração e Solução) podem ser facilmente migrados para o ambiente Unix.

Para dar continuidade ao projeto de *ALLOS*, propõem-se as seguintes sugestões de pesquisa:

- Implementação de uma interface gráfica para a especificação de modelos de redes de filas, de forma que a interação do sistema com o usuário se torne cada vez mais amigável e flexível;
- acrescentar outras medidas de desempenho ao módulo solução de *ALLOS*, de forma que o usuário disponha de mais informações para verificar o comportamento do modelo em relação às demandas solicitadas;
- estender as disciplinas de escalonamento de clientes em fila como, por exemplo, LCFS (Last Come First Served) e aleatória, tornando a ferramenta mais abrangente;
- ampliar o domínio de aplicação da ferramenta: (a) permitindo avaliar modelos com roteamento probabilístico, conforme especificado na interface do *SAVAD* [SOUTO, 1993], e (b) incluir a opção de descarte de clientes que chegam a um nodo cuja fila se encontra no seu limite de ocupação;

- estender a funcionalidade dos elementos de modelagem do tipo *estação de serviço*, de sorte que estes possam atender clientes de classes diferentes;
- aumentar a versatilidade da ferramenta, disponibilizando-a para uso em estações de trabalho, em ambiente Unix, permitindo assim, que modelos com espaço de estados maiores sejam solucionados, reduzindo a forte restrição imposta pela modelagem markoviana, que é a explosão de estados;
- expandir a capacidade do módulo solução de *ALLOS*, a fim de permitir a escolha do método de solução mais eficiente, entre os vários disponíveis na literatura [SILVA, 1992b]. Assim, o módulo solução poderá ser capaz de escolher entre um método iterativo ou direto, baseando-se na densidade da matriz de transição de estados;
- tornar, em uma outra versão, a ferramenta mais abrangente, de tal forma que, para um modelo com um espaço de estados muito grande (na ordem de 10^6 estados [SILVA, 1992b]) a ferramenta possa, automaticamente, truncar o espaço de estados do modelo, através da escolha de estados e transições a serem incluídos no modelo. Esta escolha poderá ser feita, por exemplo, em função das medidas de desempenho de interesse a serem computadas.

Abstract

The relevance of markovian models in performance evaluation of systems and the necessity of efficient tools for specification and solution of models are the aim of the new generation of concurrent systems. These tools must be sophisticated in order to make the mathematical definition of the modeled system and the details about the techniques of solution of the model transparent to their users.

This dissertation presents *ALLOS*, a friendly tool to solve models for networks of queues using Markov chains. The state space generation process for a specific model is automatically performed from a given initial state, according to the rules that describe the behavior of that model.

ALLOS is part of an intelligent and integrated simulation environment called *SAVAD*, a multidisciplinary project which involves the Computer Networks and Artificial Intelligence Groups of Federal University of Paraiba.

The use of this tool is adequate for solving systems models which exhibit resource contention, such as computer systems and computer networks and others which can be modeled using the network of queues paradigm.

Referências bibliográficas

- ARITY CORPORATION. The Arity prolog language reference manual; version 5.0. Massachusetts (USA), 1988. 322p.
- BARROSO, L. C.; BARROSO, M. M. de A.; CAMPOS FILHO, F. F.; CARVALHO, M. L. B. de; MAIA, M. L. Cálculo numérico - com aplicações. São Paulo: Harbra, 1987. 367p.
- BERSON, S.; SILVA, E. de S. e; MUNTZ, M. M. An object oriented methodology for the specification of Markov models. In: INTERNATIONAL CONFERENCE ON THE NUMERICAL SOLUTION OF MARKOV CHAINS, 1., 1990. p. 2-29.
- BIRTWISTLE, G. M. et al. Simula begin. Philadelphia: Auerbach Publisher, 1973.
- BOOCH, G. Object oriented design. s.l.: Benjamin/Cummings, 1991.
- BURKE, P. J. The output of a queueing system. In: Operations Research, v.4, p. 699-704, 1956.
- CABRAL, M. I. C. Modelagem do protocolo da camada de sessão. Campina Grande: UFPB/CCT/CCPgEE, 1987, 146p. (Tese de doutorado)
- CABRAL, M. I. C.; SOUTO, F. A. C.; CASTRO FILHO, H. C.; SILVA, H. DE M.; BRASILEIRO, M. A. G. An integrated system for modeling and evaluating models of networks of queues. In: PROCEEDINGS OF THE IASTED INTERNATIONAL CONFERENCE - MODELING AND SIMULATION - MS'94. Pittsburgh: IASTED, 1994. p. 206-209.
- CHIOLA, G. A software package for the analysis of generalized stochastic Petri net models. In: PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON TIMED PETRI-NETS, 1985. Proceedings of the international workshop on timed Petri-nets. s.l.: s.ed., 1989. p. 136-143.
- CIARDO, G. ; MUPPALA, J.; TRIVEDI, K. S. SPNP: Stochastic Petri net package. In: INTERNATIONAL WORKSHOP ON PETRI-NETS AND PERFORMANCE MODELS, 3., 1989. Proceedings of the Third International Workshop on Petri-nets and Performance Models. s.l.: s.ed., 1989, p. 142-151.
- CLOCKSIN, W. F.; MELLISH, C. S. Programming in Prolog. Berlin: Springer-Verlag, 1981.

- CONCEIÇÃO FILHO, H. C. SIM/SAVAD - Um simulador de modelos de redes de filas. Campina Grande: UFPB/CCT/COPIN, 1993. 162p (Dissertação de Mestrado).
- DIMITRIJEVIC, D. D.; CHEN, M. S. An integrated algorithm for probabilistic protocol verification and evaluation. In: PROCEEDINGS OF INFOCOM'89, 1989.
- DIMITRIJEVIC, D. D.; CHEN, M. S. Dynamic state exploration in quantitative protocol analysis. Protocol specification, testing and verification. s.l.: North-Holland, 1990, p. 327-338.
- DINIZ, M. C.; SILVA, E. de S. e. Uma ferramenta para especificação e geração de modelos markovianos. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 8º, 1990. Campinas. Anais. Campinas: UNICAMP, 1990. p. 16-36.
- DINIZ, M. C.; SILVA, E. de S. e. Especificação e geração de modelos markovianos para análise de desempenho e confiabilidade de sistemas. Revista Brasileira de Computação, v. 6, n. 3, p. 23-42, jan-mar, 1991.
- ECKEL, B. Using C++. s.l.: McGrawHill, 1989.
- GOYAL, A.; CARTER, W. C.; SILVA, E. de S. e; LAVENBERG, S.S.; TRIVEDI, K.S. The system availability estimator. In: PROCEEDINGS OF FTCS-16, 1986. p. 84-89.
- JOHNSON JUNIOR, A. M.; MALEK, M. Survey of software tools for evaluating reliability, availability and serviceability. ACM Computing Surveys, v.20, p. 227-271, 1988.
- KLEINROCK, L. Queueing systems; Theory. New York: Wiley Interscience, 1975. 417p.
- KOBAYASHI, H. Modeling and analysis: an introduction to system performance evaluation methodology. New York: Addison-Wesley, 1978. 446p.
- LITTLE, J. D. C. A proof of the queuing formula $L = \lambda W$. Operations Research, v.9, p. 383-387, 1961.
- MAKAN, S. V.; AVIZIENIS, A. ARIES 81: A reliability and life-cycle evaluation tool. In: PROCEEDINGS OF FTCS-12, 1982, p. 274-276.
- MOURA, J. A. B.; SAUVÉ, J. P.; GIOZZA, W. F.; ARAÚJO, F. M. Redes locais de computadores: Protocolos de alto nível e avaliação de desempenho. São Paulo: McGraw-Hill, 1986. 446p.
- MCCOMAS, M. G., LAW, A. M. Simulation software for communications networks: The State of the Art. IEEE Comm. Magazine, p. 44-50, March. 1994.
- OLIVEIRA, S. R. de M.; CABRAL, M. I. C.; FERNEDA, E.; BRASILEIRO, M. A. G. ALLOS - Uma ferramenta para solucionar modelos de redes de filas usando cadeias de Markov. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 13º, 1995. Belo Horizonte. Anais. Belo Horizonte: UFMG, 1995a. p. 213-225.

- OLIVEIRA, S. R. de M.; CABRAL, M. I. C.; FERNEDA, E.; BRASILEIRO, M. A. G. ALLOS - A tool to solve markovian models. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE APPLIED ON MODELLING, SIMULATION AND OPTIMIZATION. Cancún, México: IASTED, June, 1995b, 5p.
- PAGE JUNIOR., T. W.; BERSON, S. E.; CHENG, W. C.; MUNTZ, R. R. An object-oriented modeling environment. ACM Sigplan Notices, v.24, n.10, p. 287-296, 1989.
- REISER, M. A queueing network analysis of computer communications networks with window flow control. IEEE Trans. Comm., v.27, n.8, p. 1199-1209, 1979.
- REISER, M.; LAVENBERG, S. Mean value analysis of closed multichain queueing networks. JACM, v.27, p. 313-322, 1980.
- SAUER, C. H.; CHANDY, K. M. Computer systems performance modeling. Englewood Cliffs, New Jersey: Prentice-Hall, 1981. 352p.
- SAUER, C. H.; MACNAIR, E. The evaluation of the research queueing package RESQ. In: Modelling techniques and tools for performance analysis. s.l.: North-Holland. 1985, p. 5-24.
- SHRIBER, T. J. Simulation using GPSS. s.l.: John Willey & Sons, 1974.
- SILVA, E. de S. e. An object oriented methodology for the specification of Markov chains. Rio de Janeiro: UFRJ/NCE, 1988. 30p. (Relatórios técnicos).
- SILVA, E. de S. e; OCHOA, P. M. State space exploration in Markov models. Performance Evaluation Review, v.20, n.1, p. 152-166, june. 1992a.
- SILVA, E. de S. e; MUNTZ, R. R. Métodos computacionais de solução de cadeias de Markov: aplicações a sistemas de computação e comunicação. Porto Alegre: UFRGS, 1992b. 206p.
- SOUTO, F. de A. C. SAVAD - sistema de avaliação de desempenho de modelos de redes de filas. Campina Grande: UFPB/CCT/COPIN, 1993, 109p (Dissertação de Mestrado).
- STROUSTRUP, B. The C++ Programming language. Califórnia: Addison-Wesley, 1987. 328p.
- WALLACE, V. L.; ROSEMBERG, R. S. Markovian models and numerical analysis of computer system behavior. In: AFIPS CONFERENCE PROC. 28. 1966. p. 141-148.
- WIENER, R. S.; PINSON, L. J. Programação orientada para objeto e C++. São Paulo: Makron Books do Brasil, 1991. 333p.
- ZORTECH INCORPORATED. C++ Compiler version 2.1. Massachusetts, 1990. 379p.