



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-15257-TDI/1342

MODELAGEM MARKOVIANA DE COMUNICAÇÃO EM REDES DE SATÉLITES

Heitor Angelo Moraes Júnior

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelos Drs. Solon Venâncio de Carvalho e Rita de Cássia Meneses Rodrigues,
aprovada em 2 de abril de 2008.

O original deste documento está disponível em:

<<http://urlib.net/sid.inpe.br/mtc-m17@80/2008/02.12.12.07>>

INPE
São José dos Campos
2008

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:

Presidente:

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

Membros:

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

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

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

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

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

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

BIBLIOTECA DIGITAL:

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

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

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

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

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

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

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

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

EDITORAÇÃO ELETRÔNICA:

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



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-15257-TDI/1342

MODELAGEM MARKOVIANA DE COMUNICAÇÃO EM REDES DE SATÉLITES

Heitor Angelo Moraes Júnior

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelos Drs. Solon Venâncio de Carvalho e Rita de Cássia Meneses Rodrigues,
aprovada em 2 de abril de 2008.

O original deste documento está disponível em:

<<http://urlib.net/sid.inpe.br/mtc-m17@80/2008/02.12.12.07>>

INPE
São José dos Campos
2008

Dados Internacionais de Catalogação na Publicação (CIP)

M791m Moraes Júnior, Heitor Angelo.

Modelagem Markoviana de comunicação em redes de satélites/ Heitor Angelo Moraes Júnior. – São José dos Campos: INPE, 2008.

131p. ; (INPE-15257-TDI/1342)

1. Cadeia de Markov. 2. Processos estocásticos. 3. Modelagem de sistemas de satélites. 4. Método de decomposição. 5. Pesquisa operacional. I. Título.

CDU 303.092.5

Copyright © 2008 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, microfilmico, reprográfico ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2008 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

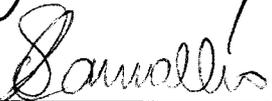
Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Mestre em
Computação Aplicada

Dr. José Demisio Simões da Silva



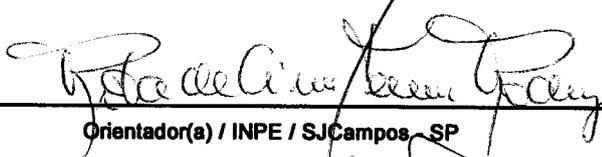
Presidente / INPE / SJC Campos - SP

Dr. Solon Venâncio de Carvalho



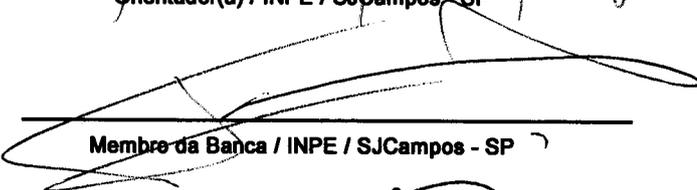
Orientador(a) / INPE / SJC Campos - SP

Dra. Rita de Cássia Meneses Rodrigues



Orientador(a) / INPE / SJC Campos - SP

Dr. Horacio Hideki Yanasse



Membro da Banca / INPE / SJC Campos - SP

Dra. Mischel Carmen Neyra Belderrain



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

Aluno (a): Heitor Angelo Moraes Junior

São José dos Campos, 02 de Abril de 2008

FOLHA DE APROVAÇÃO

CONFECCIONADA PELO SPG E INCLUÍDA PELO SID.

“Creio no supremo valor da pessoa humana, no seu direito à vida, à felicidade e à procura da felicidade.

Creio que cada direito acarreta uma responsabilidade, cada oportunidade uma obrigação, cada posse um dever.

.....

Creio em Deus todo sabedoria e todo amor, não importa como seja chamado, e que a máxima satisfação, a maior felicidade, o mais amplo sentido de realização são alcançados vivendo em harmonia com a sua vontade”.

John Davison Rockefeller Jr.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me proporcionar mais este desafio de vida, por me dar condições físicas e mentais para desenvolver este trabalho e por colocar em minha vida as pessoas que me auxiliaram em mais esta jornada.

A minha família, aos meus amigos e a todos aqueles que de alguma forma contribuíram para o êxito deste trabalho. Principalmente àqueles que mesmo sentindo a minha ausência, sempre me incentivaram, dividindo alegrias e tristezas, sem medir esforços para que eu vencesse mais esta caminhada.

Aos meus orientadores Profa. Dra. Rita de Cássia Meneses Rodrigues e Prof. Dr. Solon Venâncio de Carvalho pela pronta disposição em me orientar, pela confiança, pelo conhecimento passado, não medindo esforços, paciência e atenção necessária para atingirmos nossa meta.

Aos professores do INPE por compartilhar seus conhecimentos.

Ao Instituto Nacional de Pesquisas Espaciais – INPE – pela disponibilidade de seus recursos e instalações.

À CAPES por ter ajudado a financiar este projeto.

RESUMO

Um dos objetivos de sistemas de satélites de comunicação é conectar pessoas distantes geograficamente, permitindo a transmissão de dados, sinais de rádio e televisão, telefonia móvel e internet, entre outros. Neste trabalho modela-se um sistema de satélites de comunicação por um processo markoviano. Inicialmente, considera-se uma constelação de seis satélites geostacionários em uma única órbita. Posteriormente, considera-se que os satélites se movem em relação à Terra e, assim, os sinais, que são emitidos para um satélite, são transferidos para o próximo satélite que ocupa seu lugar, operação que é denominada *handoff*. Admite-se que as chegadas de chamadas são uniformemente distribuídas nas áreas de cobertura dos satélites e seguem uma distribuição de Poisson, e que os tempos de duração das chamadas são exponencialmente distribuídos. Expressões são apresentadas para se obter as seguintes medidas de desempenho: número médio de chamadas entre os satélites, número médio de canais usados, probabilidade de bloqueio e taxa de rejeição de chamadas no sistema e em cada conexão, e taxa efetiva de chegada de chamadas em cada conexão. No modelo markoviano proposto o espaço de estados cresce exponencialmente quando o número de satélites aumenta, portanto, apresenta-se um método de decomposição que permita tratar com sistemas geostacionários maiores. Resultados numéricos são apresentados.

MARKOV MODEL OF SATELLITE COMMUNICATION NETWORKS

ABSTRACT

One of the main purposes of satellite communication systems is to connect geographically distant individuals, allowing the transmission of data, radio and television signs, mobile telephony, internet connection, and others. In this work we model a satellites communication system by a Markov process. Initially, we consider a constellation with six geostationary satellites in a single orbit. Then, we consider that the satellites move relatively to the Earth, and, hence, the signs that are being emitted for a satellite are transferred to the next satellite that occupies its place, operation called handoff. We assume that calls arrivals are uniformly distributed over the satellites cover areas and follow a Poisson distribution, and that call duration times are exponentially distributed. Expressions are presented for obtaining the following performance measures: the average number of calls between satellites, average numbers of used channels, call-blocking probabilities and call rejection rate in each connection and in the system, and effective arrival rate of the calls in each connection. In the proposal Markov model the process' state space grows exponentially when the number of satellites increase, therefore, we present a decomposition method that allows us to handle larger geostationary systems. Numeric results are presented.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

LISTA DE SÍMBOLOS

1 INTRODUÇÃO	23
1.1 Revisão Bibliográfica.....	25
1.2 Objetivo e estrutura do texto.....	28
2 CONCEITOS BÁSICOS	31
2.1 Sistemas de satélites.....	31
2.2 Cadeia de Markov a Tempo Contínuo (CMTC).....	34
2.3 Teoria de Filas.....	37
3 MODELAGEM DO SISTEMA DE SATÉLITES	43
3.1 Sistema em estudo.....	43
3.2 O modelo markoviano.....	45
3.3 Cálculo das probabilidades estacionárias do sistema.....	48
3.3.1 Solução na Forma de Produto.....	49
3.4 Medidas de desempenho.....	52
4 SISTEMA DE SATÉLITE COM <i>HANDOFF</i>	55
4.1 Sistema em estudo.....	55
4.2 Modelagem markoviana.....	57
5 MÉTODO DE DECOMPOSIÇÃO	61
5.1 Método de decomposição.....	61
5.2 Duração média das chamadas.....	66
5.3 Algoritmo de decomposição.....	67
5.4 Probabilidade de bloqueio de chamadas nas conexões entre os dois subsistemas (ZAIM, 2001).....	70
5.5 Probabilidade de bloqueio de chamadas nas conexões entre os dois subsistemas (ZAIM ET AL., 2003).....	71
6 RESULTADOS NUMÉRICOS	73
6.1 Sistema Geoestacionário.....	73
6.2 Sistema com <i>handoff</i> ou não-geoestacionário.....	79
6.3 Sistema decomposto.....	80
6.4 Sistema com distribuição heterogênea de clientes sobre a superfície terrestre.....	83

7 CONCLUSÃO	87
7.1 Propostas de trabalhos futuros	88
REFERÊNCIAS BIBLIOGRÁFICAS	91
A APÊNDICE A – INPLEMENTAÇÃO COMPUTACIONAL	95
A.1 Código da implementação do modelo de sistema de satélites geoestacionários sem decomposição apresentado no Capítulo 3	96
A.2 Código da implementação do modelo de sistema de satélites não- geoestacionários (com <i>handoff</i>) apresentado no Capítulo 4	105
A.3 Código da implementação do modelo de sistema de satélites geoestacionários com decomposição, utilizando as equações de probabilidade de bloqueio entre subsistemas sugeridas por Zaim (2001); Zaim et al. (2003); e sugeridas por este trabalho, apresentadas no Capítulo 5	118

LISTA DE FIGURAS

2.1	Quatro satélites em uma órbita única	34
2.2	Elementos de um sistema de filas	38
2.3	Diagrama de transição de estados para sistema de fila M/M/c/c.....	40
3.1	Seis satélites em uma órbita única	45
4.1	Sistema de seis satélites em órbita baixa	56
4.2	Itens da modelagem do <i>handoff</i>	57
5.1	Sistema de seis satélites decomposto	62
5.2	Algoritmo de decomposição	69
6.1	Variação da probabilidade de bloqueio do sistema em função da taxa de chegada de chamadas.....	75
6.2	Variação da probabilidade de bloqueio em cada conexão quando a quantidade de canais nas conexões ISL é maior	77
6.3	Variação da probabilidade de bloqueio em cada conexão quando a quantidade de canais nas conexões UDL é maior	78
6.4	Probabilidade de bloqueio em cada conexão do sistema para cada método de decomposição, considerando $C_{ISL}=C_{UDL}=2$	82
6.5	Probabilidade de bloqueio em cada conexão do sistema para cada método de decomposição, considerando $C_{ISL}=C_{UDL}=3$	83

LISTA DE TABELAS

2.1	Tipos de órbitas de satélites	33
4.1	Transições devido ao <i>handoff</i>	59
6.1	Número de estados e transições	74
6.2	Número médio de chamadas entre dois satélites	76
6.3	Número médio de canais usados nas conexões físicas	76
6.4	Variação da probabilidade de bloqueio do sistema em função da taxa de <i>handoff</i>	79
6.5	Probabilidade de bloqueio em um sistema de seis satélites sem e com decomposição, com $C_{ISL}=C_{UDL}=2$	80
6.6	Probabilidade de bloqueio em um sistema de seis satélites sem e com decomposição, com $C_{ISL}=C_{UDL}=3$	81
6.7	Taxas de chegada de chamadas em cada conexão para uma distribuição heterogênea de clientes sobre a superfície terrestre	84
6.8	Probabilidade de bloqueio de chamadas considerando rotas diferentes...	85

LISTA DE SIGLAS E ABREVIATURAS

MECB	Missão Espacial Completa Brasileira
CMTC	Cadeia de Markov a Tempo Contínuo
M/M/c/c	Sistema de filas em que o tempo entre chegadas e a duração dos serviços (duração das chamadas) são exponencialmente distribuídos, existem c servidores no sistema para atender os clientes (c canais) e a capacidade do sistema é limitada a c clientes (chamadas)
M/M/c	Sistema de filas em que o tempo entre chegadas e a duração do serviço (duração das chamadas) são exponencialmente distribuídos, existem c servidores no sistema para atender os clientes (c canais) e a capacidade do sistema é ilimitada
LEO	<i>Low Earth Orbit</i>
MEO	<i>Medium Earth Orbit</i>
GEO	<i>Geostationary Earth Orbit</i>
ISL	<i>Inter Satellite Link</i>
UDL	<i>Up-and-Down Link</i>
FCFS	<i>First Come, First Served</i>

LISTA DE SÍMBOLOS

i, j	Satélite
$C_{UDL i-i}$	Número de canais UDL abaixo do satélite i
C_{UDL}	Número de canais UDL
$C_{ISL i-j}$	Número de canais ISL entre o satélite i e o satélite j
C_{ISL}	Número de canais ISL
E	Espaço de estados
X_t	Processo estocástico com parâmetro t
x_n	Estado do sistema no instante n
e, x, y, \hat{e}, e_h	Estado do sistema
t, n, s	Instante de tempo
T	Espaço de parâmetros
$Q(x_n, x_{n+1})$	Probabilidade de transição do estado x_n para o estado x_{n+1}
Q	Matriz de transição da cadeia de Markov
$\lambda(x)$	Taxa de transição da cadeia de Markov a tempo contínuo
$P_t(x, y)$	Função de transição da cadeia de Markov a tempo contínuo
$\pi(y)$	Probabilidade limite do estado y
λ	Taxa de chegada de clientes na fila
μ	Taxa de serviço, Taxa de duração das chamadas
λ_{ij}	Taxa de chegada de chamadas entre o satélite i e j
μ_{ij}	Taxa de término de chamadas entre o satélite i e j
ρ	Densidade de tráfego
n_{ij}	Número de chamadas bidirecionais entre o satélite i e o satélite j
$n_{ij}(e)$	Número de chamadas entre os satélites i e j quando o sistema está no estado e
k	Número de satélites do sistema
N	Quantidade de conexões lógicas do sistema
$r(e, \hat{e})$	Taxa de transição do estado e para o estado \hat{e}
G	Constante de normalização
$\bar{X}(n_{ij})$	Número médio de chamadas entre os satélites i e j

$\overline{N_{UDLi}}$	Número médio de canais utilizados em cada conexão UDL abaixo do satélite i
$\overline{N_{ISLi-j}}$	Número médio de canais utilizados em cada conexão ISL entre o satélite i e o satélite j
π_b	Probabilidade de bloqueio de chamadas do sistema
$\pi_b(i, j), p_{ij}$	Probabilidade de bloqueio de chamadas na conexão $i-j$
p'_{ij}	Probabilidade de bloqueio de chamadas na conexão $i-j$ do subsistema 1
p''_{ij}	Probabilidade de bloqueio de chamadas na conexão $i-j$ do subsistema 2
B_{ij}	Conjunto de estados que não podem receber chamadas
θ	Taxa de rejeição de chamadas do sistema
θ_{ij}	Taxa de rejeição de chamadas na conexão $i-j$
f_{ij}	Probabilidade de que uma chamada originada na área de cobertura do satélite i seja para um cliente na área de cobertura do satélite j .
A	Área de cobertura de um satélite
v	Velocidade do satélite em relação a superfície terrestre
v_A	Velocidade da área de cobertura do satélite
Δt	Intervalo de tempo
ΔL	Distância que o satélite percorre no intervalo de tempo Δt
ΔA	Parcela da área de cobertura do satélite que foi transferida para o próximo satélite
β	Fator de forma que depende do formato da área de cobertura do satélite
g	Probabilidade que uma chamada seja transferida para o próximo satélite dentro do intervalo de tempo Δt

1 INTRODUÇÃO

O desenvolvimento de sistemas de satélites facilitou a troca de informações entre vários pontos distantes geograficamente sobre a superfície terrestre com qualidade e custo acessível. Nas últimas décadas o mundo tem acompanhado o aumento do uso de satélites para transmissão de dados, seja na área de telefonia fixa ou móvel, transmissão de sinais de rádio e televisão, coleta de dados e vários outros usos. Apesar de, nos centros urbanos, a fibra ótica estar substituindo os satélites de comunicação, em áreas rurais a comunicação por satélites ainda é, em alguns casos, a única viável.

Um dos primeiros artigos sobre o uso de satélites de comunicação foi escrito por Clarke (1945). Neste artigo o autor sugere o uso de três satélites geoestacionários, ou seja, com velocidade angular igual à velocidade de rotação da Terra, igualmente espaçados ao redor do equador, para cobrir toda a superfície terrestre. A partir da publicação de Clarke as pesquisas sobre satélites para transmissão de sinais se intensificaram e o uso de satélites nas áreas de pesquisas científicas e transmissão de sinais aumentaram. A evolução do uso de satélites de comunicação pode ser verificada através do breve histórico apresentado a seguir.

Em 1954, John R. Pierce, depois de avaliar as vantagens tecnológicas e as perspectivas financeiras de comunicação por satélites, faz um discurso e publica um artigo, em 1955, apresentando a vantagens das comunicações por rebatimento de sinais, ou comunicação em forma de espelho, para constelações de satélites em órbitas de altitude média e órbitas geoestacionárias¹. Depois do lançamento do Sputnik 1, em 1957 pela União Soviética, a NASA dedicou grande parte de seus recursos financeiros ao estudo da comunicação entre satélites através de espelhos enquanto que o Departamento de Defesa Americano investiu suas pesquisas na ampliação de sinais de comunicação nas conexões entre satélites, melhorando a qualidade das comunicações. Depois disso, várias tentativas de projetos para desenvolver e lançar satélites de comunicação foram realizadas. A partir de 1958, acumularam-se sucessos nas atividades espaciais; satélites

¹ < <http://www.hq.nasa.gov/office/pao/History/satcomhistory.html> > Acesso em: 07 abril 2008.

militares de reconhecimento, satélites científicos e operacionais e sondas espaciais cada vez mais sofisticados foram lançados ao espaço, permitindo grandes saltos no conhecimento da Terra e do Universo, e a criação e o fortalecimento de novos serviços. Em 1960 foi lançado o primeiro satélite de comunicações por reflexão – ECHO; em 1963, o primeiro satélite geoestacionário – SYNCOM; em 1965, o primeiro satélite geoestacionário comercial *Early Bird* – INTELSAT I; em 1976, foram lançados três satélites para comunicações marítimas – MARISAT; em 1982, o primeiro satélite para comunicações móveis - INMARSAT-A; em 1988, o primeiro satélite para comunicações móveis e transmissão de dados – INMARSAT-C².

Com a telefonia celular, uma nova tecnologia de sistemas para comunicação pessoal (PCS – *Personal Communications System*), que usa constelações de satélites para transmitir voz, imagem e dados a qualquer lugar, é desenvolvida. O mais ambicioso desses sistemas é o LEO Iridium, patrocinado pela Motorola. Ele possui 66 satélites em órbita polar a altitudes de aproximadamente 780 quilômetros, com seis planos orbitais separados por 30 graus em torno do equador, cada um com 11 satélites igualmente espaçados em seu plano orbital³.

Hoje, os satélites artificiais fazem parte do cotidiano das pessoas, quer por ocuparem sistematicamente os meios de comunicação, quer por proverem recursos que se tornaram indispensáveis à sociedade. O uso de satélites artificiais, permitindo a comunicação entre pessoas através de internet, telefonia celular, rádio e televisão, em qualquer parte do mundo, mudou a forma com que as pessoas passaram a interagir tanto no meio profissional como pessoal.

Vários autores têm se dedicado à modelagem e análise do desempenho das redes de satélites. O desempenho do serviço de comunicação oferecido por uma rede de satélites depende de fatores, tais como: altitude da órbita e capacidade de transmissão de informações dos satélites.

² <<http://www.aeb.gov.br>> Acesso em: 01 outubro 2007.

³ <<http://iridium.com/about/howitworks.php>> Acesso em: 28 abril 2008.

No presente trabalho, a comunicação em constelações de satélites é modelada pela quantidade de chamadas em cada uma de suas conexões (canais ligando os satélites, e canais ligando os satélites e a Terra). Considera-se que as chegadas de chamadas em cada conexão e a duração destas chamadas são aleatórias, e que a quantidade de chamadas no próximo intervalo de tempo só depende das chamadas que estão ocorrendo no instante atual e não das chamadas que já ocorreram no passado. Processos markovianos têm sido aplicados para descrever sistemas com estas características, em que o estado futuro do sistema obedece a uma distribuição de probabilidade, que é determinada pelo estado imediatamente precedente (estado atual) do sistema, independente dos seus estados passados. No presente trabalho, utiliza-se cadeia de Markov a tempo contínuo (CMTC) para modelar os sistemas de satélites em estudo. CMTC são processos markovianos cujos tempos de permanência em cada estado são exponencialmente distribuídos.

1.1 Revisão bibliográfica

O funcionamento de sistemas de satélites é objeto de estudo de vários autores. Para um estudo mais aprofundado sobre este assunto destacam-se os trabalhos de Ha (1990); Maral e Bousquet (1993); Bem et al. (2000); Sheriff e Hu (2001); Wood (2001); Elbert (2004). Ha (1990) descreve o funcionamento de satélites de comunicação em geral e analisa uma rede de satélites com antenas pequenas, denominados de VSAT (*Very Small Aperture Terminal*). Maral e Bousquet (1993) apresentam uma introdução sobre as técnicas e tecnologias de propagação de ondas de rádio, antenas e processamento de sinais em sistemas de satélite de comunicação. Bem et al. (2000) descrevem tipos de frequências, tipos de órbita, arquiteturas e problemas encontrados no uso de sistemas de satélites. Sheriff e Hu (2001) descrevem alguns sistemas de comunicação para telefonia móvel, apresentando conceitos básicos de sistemas de telefonia móvel, tipos de órbita, arquitetura de redes, alguns sistemas reais, funcionamento dos canais de comunicação e uma análise de mercado. Wood (2001) analisa o uso de redes de satélites para acessar internet, sua topologia, as vantagens e desvantagens do uso de diferentes tipos de órbitas, atrasos por causa de tráfego e o

roteamento e fluxo de informações nas conexões entre satélites. Elbert (2004) apresenta um manual sobre satélites de comunicação, mostrando características de seu funcionamento como múltiplos acessos, tipos de frequência, sinais digitais, arquiteturas de sistemas de satélites, com enfoque para aspectos técnicos e econômicos dos sistemas e dos serviços utilizados por empresas e usuários.

Para uma introdução aos processos de Markov destacam-se Çinlar (1975); Ross (1982); Stewart (1994). Çinlar (1975) apresenta uma introdução aos principais processos estocásticos, incluindo exemplos e exercícios que auxiliam no entendimento da teoria apresentada. Ross (1982) acrescenta, aos processos descritos em Çinlar, os movimentos brownianos, caminhos aleatórios e martingales. Stewart (1994) se concentra nos processos markovianos, dando ênfase ao tratamento detalhado das soluções numéricas destes processos, especialmente quando o espaço de estado é muito grande. Neste contexto, ele considera métodos diretos, recursivos e iterativos de agregação e desagregação, Ele também discute cadeias de Markov periódicas e redes autômatas estocásticas.

Entre os vários trabalhos que utilizam uma modelagem markoviana para sistemas de satélites destacam-se Altman (2000); Usaha e Barria (2002); Sumter (2003). Altman (2000) apresenta uma ampla revisão bibliográfica sobre aplicações de processos de decisão markovianos a redes de comunicação. Nesta referência diversos problemas encontrados em redes de comunicação, tais como controle de fluxo e congestionamento de rede, são analisados. Usaha e Barria (2002) estudam o problema de admissão de chamadas e seleção de rotas em redes de satélites do tipo LEO (*Low Earth Orbit*). Eles propõem a utilização de modelos semi-markovianos de decisão para obter políticas de roteamento e admissão de chamadas com o objetivo de maximizar o rendimento da rede a longo prazo, que é definido como uma combinação entre o rendimento ganho ao se aceitar novas chamadas e o custo do término forçado de chamadas em andamento. Sumter (2003) apresenta um modelo markoviano de decisão a tempo discreto de uma constelação de satélites, com o objetivo de encontrar políticas ótimas de reposição de satélites que minimize o custo esperado total de manutenção desta constelação. O custo

associado a cada ação é determinado pelo número de satélites adquiridos, lançados ou mantidos em estoque e pela capacidade operacional da constelação. O autor utiliza um algoritmo de iteração de valores (PUTTERMAN, 1994) para determinar a política de substituição ótima e o custo esperado total mínimo.

A modelagem markoviana está intimamente ligada à Teoria de Filas. Para saber mais sobre processos de Markov e Teoria de Filas destacam-se Allen (1978); Cooper (1981); Asmussen (1987); Gelenbe e Pujolle (1987); Wolff (1989); Bolch et al. (1998). Allen (1978) aborda o tema com ênfase em estatística e probabilidade com aplicações a ciência da computação. Cooper (1981) apresenta modelos de filas uni e multidimensionais e simulações para alguns modelos. Asmussen (1987) apresenta uma introdução à cadeia de Markov, teoria de filas, teoria de renovação, processos de regeneração e passeios aleatórios (*Random Walks*). Gelenbe e Pujolle (1987) apresentam um estudo sobre filas com um único servidor, algumas extensões dessas filas e uma pequena abordagem a redes de filas. Wolff (1989) apresenta um estudo mais detalhado sobre vários tipos de filas e sobre como tratar redes de filas, e Bolch et al. (1998) apresentam um tratamento detalhado de sistemas de filas, redes de filas, e cadeias de Markov a tempo discreto e contínuo. Além disso, os autores descrevem alguns métodos de aproximação para tratar estes sistemas.

O presente trabalho foi baseado no trabalho de Zaim (2001). Nesta referência, o autor apresenta um modelo markoviano para constelações de satélites geoestacionários em uma única órbita. Além disso, o autor apresenta um método de aproximação para calcular probabilidades de bloqueio em constelações com um grande número de satélites. O método é baseado na decomposição do sistema em subsistemas com poucos satélites. Cada subsistema é analisado separadamente, e suas probabilidades limites são calculadas através da solução na forma de produto. Os resultados obtidos de cada subsistema são combinados de forma iterativa de maneira a resolver todo o sistema. Este método é, então, generalizado para uma constelação de satélites, que se movem em relação à superfície terrestre, envolvendo muitas órbitas, e, assim, considerando operações de *handoff* (operação de transferência de chamadas de um satélite para outro).

Finalmente, o autor desenvolve um método para calcular limites superiores e inferiores para as probabilidades de bloqueio de chamadas.

1.2 Objetivo e estrutura do texto

O objetivo deste trabalho é modelar e analisar o desempenho da comunicação em rede de satélites.

Inicialmente, baseando-se no trabalho de Zaim (2001), modela-se uma constelação de seis satélites de comunicação em uma única órbita geoestacionária, utilizando-se processos markovianos, mais especificamente cadeia de Markov a tempo contínuo. O sistema é, então, decomposto em subsistemas de três satélites, utilizando-se o método de decomposição proposto por Zaim (2001). Finalmente, modela-se um sistema de satélites no qual os satélites se movem em uma velocidade angular de rotação maior que a velocidade da Terra. Neste caso, os sinais que estão sendo emitidos para um satélite são transferidos para o próximo satélite que ocupa o seu lugar, ocorrência que é denominada *handoff*.

As contribuições contidas no presente trabalho incluem, novas medidas para análise do desempenho do sistema (número médio de chamadas e número médio de canais utilizados em cada conexão do sistema, taxa de rejeição e taxa de atendimento do sistema), novas expressões para o cálculo das taxas de duração de chamadas e para o cálculo das probabilidades de bloqueio no sistema decomposto.

A estrutura do trabalho é a seguinte.

No Capítulo 2 são apresentados conceitos básicos utilizados neste trabalho. Na Seção 2.1 descrevem-se algumas características do funcionamento de sistemas de satélites. Na Seção 2.2 e 2.3 apresentam-se, respectivamente, conceitos básicos de cadeia de Markov a tempo contínuo e de teoria de filas.

No Capítulo 3 apresenta-se a modelagem do sistema de satélites Geoestacionários. Na Seção 3.1 descreve-se o sistema de seis satélites considerado neste trabalho. Na Seção 3.2 e 3.3, apresentam-se, respectivamente, o modelo markoviano proposto para este sistema e o cálculo das probabilidades estacionárias da CMTC considerada. E na Seção 3.4 apresentam-se equações para o cálculo de medidas de desempenho do sistema.

No Capítulo 4 remove-se a suposição de que a constelação é fixa sobre a superfície terrestre, admitindo-se operações de *handoff*. Na Seção 4.1 descreve-se o novo sistema em estudo e na Seção 4.2 o modelo markoviano proposto para este sistema.

No Capítulo 5 apresenta-se um método de decomposição para tratar sistemas com um grande número de satélites. Na Seção 5.1 decompõe-se a constelação de satélites em estudo em dois subsistemas menores. Na Seção 5.2, desenvolvem-se expressões para obtenção das taxas de duração de chamadas em cada subsistema. Na Seção 5.3 apresenta-se o algoritmo de decomposição desenvolvido por Zaim (2001) para obtenção das probabilidades de bloqueio do sistema decomposto, e propõem-se novas expressões para o cálculo destas probabilidades. Na Seção 5.4 e 5.5 são descritas, respectivamente, as equações, propostas por Zaim (2001) e Zaim et al. (2003), para o cálculo das probabilidades de bloqueio entre satélites que estão em subsistemas diferentes.

No Capítulo 6 apresentam-se alguns resultados numéricos, e no Capítulo 7 as conclusões deste trabalho e extensões para trabalhos futuros.

Os códigos da implementação computacional, em linguagem C++, dos modelos, que são descritos nos Capítulos 3, 4 e 5, são apresentados no Apêndice A, no final deste trabalho.

2 CONCEITOS BÁSICOS

Neste capítulo, são introduzidos alguns conceitos básicos que serão utilizados na modelagem do sistema de satélites em estudo. Na próxima seção são descritas algumas características do funcionamento das constelações de satélites. Na seção 2.3 definem-se as propriedades básicas das Cadeias de Markov a Tempo Contínuo, que serão utilizadas para modelar o sistema de satélites em estudo neste trabalho. Na Seção 2.4 faz-se uma breve introdução à Teoria de Filas, pois se pretende representar a constelação de satélites como uma rede de filas.

2.1 Sistemas de satélites

Os sistemas de satélites são classificados de acordo com a altitude em que estão posicionados em relação à superfície terrestre. Segundo Elbert (2004) os sistemas mais utilizados são os de baixa órbita, denominados LEO (*Low Earth Orbit*), os de órbita média, denominados MEO (*Medium Earth Orbit*), e os de órbita alta ou geoestacionários, denominados GEO (*Geostationary Earth Orbit*).

A seguir é apresentada uma breve descrição desses sistemas, segundo Wood (2001).

O sistema LEO localiza-se entre 500 e 2000 km de altitude e, por estar em tão baixa altitude, a área de cobertura de um satélite deste sistema é de, no máximo, 5% da área da superfície terrestre, sendo, portanto, necessário, um grande número de satélites para cobrir toda a superfície terrestre. As vantagens de se utilizar este tipo de satélite são a possibilidade de fazer sua manutenção, ou até mesmo reposição, em caso de defeito, e o seu período de latência (tempo que o sinal leva para percorrer a distância entre estes satélites e a Terra), que é menor que 15 ms. Satélites do sistema LEO se movem rapidamente em relação à superfície terrestre, com velocidade superior a 25.000 km/h. Eles passam sobre um determinado ponto fixo da Terra a cada duas horas, e, por isso, as chamadas que saem da sua área de cobertura são transferidas para o próximo satélite, operação que é denominada *handoff*.

O sistema MEO localiza-se entre 9000 e 11000 km de altitude. Neste sistema, os satélites possuem uma área de cobertura grande, em torno de 24% da superfície terrestre, e o tempo que o sinal leva para percorrer a distância entre estes satélites e a Terra é por volta de 40 ms. Estes satélites movem-se mais lentamente em relação à superfície terrestre, se comparados com os satélites do sistema LEO, mas operações de *handoff* também ocorrem neste sistema.

O sistema GEO localiza-se a aproximadamente 36.000 km de altitude sobre a linha do equador e sua velocidade angular é igual à velocidade de rotação da Terra, o que faz com que seus satélites pareçam estacionários em relação à superfície terrestre, daí a denominação geostacionários. Utilizando-se este sistema, é possível praticamente cobrir toda a superfície terrestre na região da linha do equador com apenas três satélites, exceto uma pequena calota em cada pólo (CLARKE, 1945). Por este sistema estar mais afastado do planeta Terra, o tempo que o sinal leva para percorrer a distância entre ele e a Terra é maior; ele exige maior potência de transmissão e receptores mais sensíveis.

Para que os satélites tenham uma vida longa, as altitudes das órbitas dos satélites devem evitar, no caso de altitudes mais baixas, o arrasto provocado pela atmosfera, que se rarefaz à medida que a altitude aumenta, e, em altitudes maiores, os cinturões de elevada radiação que rodeiam a Terra (cinturões de Van Allen). Estes cinturões possuem radiação máxima entre 1000 e 6000 quilômetros e entre 15000 e 20000 quilômetros⁴.

Como pode ser observado na Tabela 2.1, cada um dos sistemas descritos possui características bem definidas e diferenciadas de altitude de órbita, área de cobertura, posicionamento em relação à superfície terrestre e tempo de transmissão de sinais até o planeta Terra, atendendo a certo tipo de especificação de serviço de acordo com a necessidade da transmissão de sinais.

⁴ < <http://www.aeb.gov.br/conteudo.php?ida=28&idc=114> > Acesso em: 15 maio 2008.

Tabela 2.1 – Tipos de órbitas dos satélites

Órbita	Altitude (km)	Área terrestre de cobertura	Posição relativa à	Período de Órbita	Período de Latência
GEO	36.000	34%	Fixo	24h	0,25
MEO	9.000 a 11.000	24%	Móvel	6h	0,040
LEO	500 a 2.000	2,5% a 5%	Móvel	1h30m a 2h	0,005 a 0,015

A comunicação entre os satélites acontece através de um circuito de transmissão de mensagens que liga cada satélite ao seu vizinho e por onde são transmitidos os sinais. Esta conexão entre satélites é denominada *intersatellite link*, ou conexão ISL. Se a conexão ISL for entre satélites de mesma órbita a conexão é denominada *intraplane ISL* e se a conexão for entre satélites de órbitas diferentes ela é denominada *interplane ISL*.

A comunicação entre um satélite e a Terra é denominada *uplink* se a conexão origina-se da Terra para o satélite e *downlink* se a conexão origina-se do satélite para a Terra. Assim, a conexão entre um satélite e a Terra, ou vice-versa, é denominada *up-and-down link* ou conexão UDL. A transmissão de dados na mesma área de cobertura do satélite é uma transmissão bidirecional UDL. Cada satélite i possui capacidade de suportar até $C_{UDL\ i-i}$ canais bidirecionais nas conexões do tipo UDL e $C_{ISL\ i-j}$ canais bidirecionais nas conexões do tipo ISL com cada satélite $j, j \neq i$, adjacente a ele.

As conexões existentes entre satélites adjacentes e entre os satélites e a Terra são chamadas de conexões físicas. Cada conexão física possui uma quantidade pré-estabelecida de canais para transmitir dados.

Chamadas entre pontos da rede em que não existe uma conexão física são realizadas através de conexões lógicas. As conexões lógicas usam as conexões físicas para realizar a comunicação entre dois satélites. Por exemplo, suponha uma constelação com quatro satélites (denominados 1, 2, 3 e 4), como apresentada na Figura 2.1, de maneira que

existam somente as seguintes conexões físicas entre satélites: 1-2, 2-3, 3-4 e 1-4. Uma chamada entre os satélites 1 e 3, ou seja, uma conexão lógica entre esses satélites, usará, por exemplo, as conexões físicas 1-2 e 2-3 para realizar a comunicação.

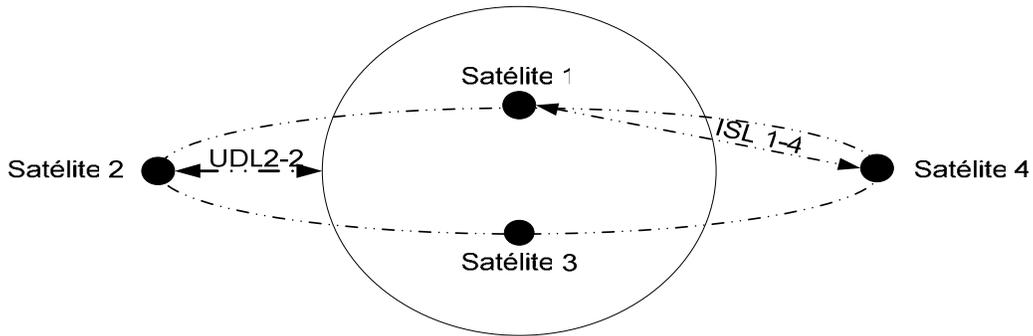


Figura 2.1 - Quatro satélites em uma órbita única.

2.2 Cadeia de Markov a Tempo Contínuo (CMTC)

Um processo estocástico com espaço de estado E é uma coleção $\{X_t; t \in T\}$ de variáveis aleatórias X_t , definidas no mesmo espaço de probabilidade e assumindo valores em E . O conjunto T é denominado seu conjunto de parâmetro. Se T é contável, especialmente se $T = \mathbb{N} = \{0, 1, \dots\}$, o processo é dito ser um processo a parâmetro discreto. Caso contrário, o processo é dito ter parâmetro contínuo. Neste caso, os exemplos mais comuns são $T = \mathbb{R}_+ = [0, \infty)$ e $T = [a, b] \subset \mathbb{R} = (-\infty, \infty)$. Geralmente o índice t representa tempo e, neste caso, X_t é o estado ou a posição do processo no tempo t (ÇINLAR, 1975).

Uma classe importante dos processos estocásticos são os processo markovianos. Nestes processos, o futuro é condicionalmente independente do passado dado que o presente é conhecido.

Se o processo markoviano tem espaço de estados E e de parâmetros T contáveis, ele é denominado Cadeia de Markov. Mas se seu espaço de parâmetros é contínuo, ele é denominado Processo de Markov ou Cadeia de Markov a tempo contínuo.

Os conceitos apresentados a seguir podem ser encontrados em Çinlar (1975).

O processo estocástico $X = \{X_n; n \in \mathbb{N}\}$ é denominado Cadeia de Markov com espaço de estados E se

$$P\{X_{n+1} = x_{n+1} \mid X_0, X_1, \dots, X_n\} = P\{X_{n+1} = x_{n+1} \mid X_n\}$$

para todo $x_{n+1} \in E$ e $n \in \mathbb{N}$.

Se a cadeia for independente de n ela é denominada homogênea no tempo, e tem-se que:

$$P\{X_{n+1} = x_{n+1} \mid X_n = x_n\} = P(x_n, x_{n+1})$$

As probabilidades $P(x_n, x_{n+1})$ são denominadas probabilidades de transição da cadeia de Markov X , e é usual apresentar seus valores em forma de matriz quadrada Q , denominada matriz de probabilidade de transição do processo.

$$Q = \begin{bmatrix} P(0,0) & P(0,1) & P(0,2) & \dots \\ P(1,0) & P(1,1) & P(1,2) & \dots \\ P(2,0) & P(2,1) & P(2,2) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

em que $P(x_n, x_{n+1}) \geq 0$, para todo $x_n, x_{n+1} \in E$ e $\sum_{x_{n+1}=0}^{+\infty} P(x_n, x_{n+1}) = 1$, para todo $x_n \in E$.

Um processo estocástico $Y = \{Y_t; t \in \mathfrak{R}_+\}$ é uma cadeia de Markov a tempo contínuo com espaço de estados E se, para qualquer $t, s \geq 0$ e $y \in E$, tem-se que:

$$P\{Y_{t+s} = y \mid Y_u; u \leq t\} = P\{Y_{t+s} = y \mid Y_t\} \quad (2.1)$$

A equação (2.1) expressa a propriedade markoviana das Cadeias de Markov a tempo Contínuo. Como a distribuição exponencial é a única distribuição contínua que têm a propriedade da falta de memória, tem-se que os tempos de permanências em uma CMTC são exponencialmente distribuídos.

Quando a probabilidade condicional (2.1) for independente de $t \geq 0$ para todo $x, y \in E$ e $s \geq 0$, a cadeia Y é denominada homogênea no tempo e a equação (2.1) pode ser escrita como:

$$P\{Y_{t+s} = y \mid Y_t = x\} = P_s(x, y)$$

Para $x, y \in E$ fixos, a função $t \rightarrow P_t(x, y)$ é denominada uma função de transição, e a família de matrizes $P_t, t \geq 0$, de funções de transição $P_t(x, y)$ é denominada função de transição da Cadeia de Markov a tempo contínuo. Por todo o texto, a análise será limitada às CMTC homogêneas.

Pode-se analisar uma CMTC através dos seus instantes de transição T_0, T_1, T_2, \dots e dos sucessivos estados visitados nestes instantes X_0, X_1, X_2, \dots . Se $X_n = x_n$, o intervalo $[T_n, T_{n+1}]$ é o intervalo de permanência em i . O processo $\{X_n, n \in \mathbb{N}\}$ é uma cadeia de Markov, denominada cadeia de Markov embutida, e $T_{n+1} - T_n$ tem distribuição exponencial com parâmetro que depende de X_n .

Teorema: Para qualquer $x, y \in E$ a função $t \rightarrow P_t(x, y)$ é diferenciável e sua derivada é contínua. Em $t = 0$, a derivada é

$$A(x, y) = \begin{cases} -\lambda(x) & \text{se } x = y, \\ \lambda(x) Q(x, y) & \text{se } x \neq y \end{cases}$$

em que Q é a matriz de transição de X , $\lambda(x)$ é o parâmetro da distribuição exponencial do tempo de permanência em x , também denominada taxa de transição da CMTC, e A é denominada matriz infinitesimal da CMTC.

Seja $\pi(y)$ a probabilidade de o sistema estar no estado y no tempo $t \rightarrow \infty$, $\pi(y) = \lim_{t \rightarrow \infty} P\{Y_t = y\}$, denominada probabilidade limite, então $\pi = (\pi(1), \pi(2), \dots, \pi(j))$ é o vetor de probabilidades limites do sistema.

Se Y é uma CMTC recorrente ergódica, ou seja, sua cadeia de Markov embutida é irredutível (conjunto formado por estados que se comunicam entre si, mas não se comunicam com nenhum estado de fora do conjunto) e recorrente positiva (se todos os estados podem ser alcançados em um número finito de passos), o sistema de equações $\mu A = 0$ tem solução única μ que é estritamente positiva. A probabilidade limite $\pi(y)$ é então dada por:

$$\pi(y) = \frac{\mu(y)}{\sum \mu(y)}, y \in E$$

Para um estudo mais aprofundado sobre processos estocásticos e, mais especificamente, processos de Markov, consultar, por exemplo, Melsa (1973); Çinlar (1975); Ross (1982); Tijms (1994); Stewart (1994).

2.3 Teoria de Filas

Considere um sistema no qual usuários se deslocam até servidores para obter algum tipo de serviço. Se o servidor está disponível, este imediatamente atende o cliente, caso contrário, o cliente entra em uma fila de espera ou deixa o sistema. Tais modelos frequentemente são tratados como filas e podem ser definidos em termos de três características: a chegada de clientes no processo, o mecanismo de serviço e a disciplina da fila.

A chegada de clientes no processo descreve a forma de solicitação de serviços. Ela é frequentemente especificada em termos da distribuição do tempo entre instantes de chegadas consecutivas de clientes. O mecanismo de serviço inclui características como o número de servidores e o tempo que o cliente utiliza o serviço do servidor. E a disciplina da fila especifica a ordem em que os usuários são selecionados da fila para serem atendidos.

A Figura 2.2 apresenta os elementos de um sistema de filas.

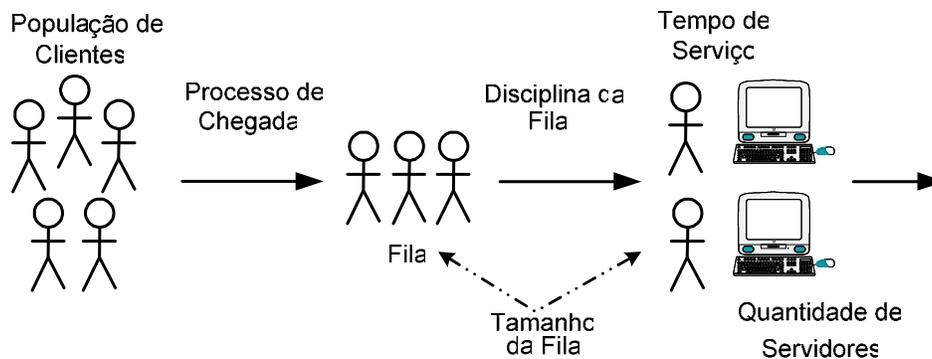


Figura 2.2 – Elementos de um sistema de filas.

Fonte: Adaptada de Pinheiro (2004).

População de Clientes é a quantidade de clientes em potencial que podem, em algum momento, decidir usar um servidor. Esses clientes chegam em instantes t_1, t_2, \dots, t_j no tempo, e as variáveis aleatórias $s_j = t_j - t_{j-1}$ representam os intervalos de tempo entre chegadas. Em geral, admite-se que não mais do que um cliente pode chegar ao sistema no mesmo instante de tempo; que o processo de chegada não varia ao longo do tempo (processo estacionário); e que ele não é afetado pelo número de usuários presentes no sistema. À medida que os clientes chegam ao sistema, eles vão sendo atendidos de acordo com a Disciplina da Fila. A mais comum é a FCFS (*first come first served*), o primeiro cliente a chegar é o primeiro a ser atendido. Mas pode-se ter outras disciplinas, tais como: LCFS (*last come, first served*), último a chegar; primeiro a ser atendido, SIRO (*service in random order*) serviço em ordem aleatória; seleção em alguma ordem de prioridade, como tempo esperado de atendimento (menos demorado primeiro), idade

do cliente (os idosos primeiro), maior sensibilidade a atrasos (mais sensíveis primeiro), qualidade de serviço, entre outros. A capacidade de clientes no sistema poderá ser limitada por questões de espaço ou para evitar um tempo de espera muito longo. O Tamanho da Fila inclui os clientes esperando atendimento e os que estão sendo atendidos. O tempo gasto por cada cliente sendo atendido pelo servidor é denominado Tempo de Serviço. Em geral, admite-se que não mais de um usuário pode ser atendido por um servidor no mesmo instante; que o processo de serviço não varia ao longo do tempo (processo estacionário) e que ele não é afetado pelo número de usuários no sistema. A Quantidade de Servidores em um sistema de filas pode variar entre: um único servidor que atende a apenas um cliente de cada vez; vários servidores, por exemplo, c servidores atendendo c clientes simultaneamente; ou uma quantidade infinita de servidores, desta forma cada cliente que chega encontra sempre um servidor disponível.

Um sistema de fila é classificado de acordo com as características de seus elementos. Normalmente utiliza-se a notação de Kendall para sua classificação: $A / B / c / K / m / Z$ onde:

A = distribuição do tempo entre chegadas;

B = distribuição do tempo de serviço;

c = número de servidores;

K = número máximo de clientes permitidos no sistema ou capacidade do sistema;

m = tamanho da população;

Z = disciplina da fila.

Usualmente, uma notação mais curta ($A / B / c$) é utilizada. As distribuições utilizadas para o tempo entre chegadas e o tempo de serviço são simbolizadas da seguinte forma:

M = Exponencial – markoviana, sem memória;

E_k = Erlang com parâmetro k ;

D = Determinística;

G = Genérica;

GI = Genérica com tempo entre chegadas independente;

H_k = Hiperexponencial com parâmetro k .

Como exemplo, pode-se citar a fila markoviana infinita M/M/c, em que as chegadas seguem um processo de Poisson, ou seja, os intervalos de tempo entre chegadas têm distribuição exponencial; os tempos de serviço são exponencialmente distribuídos; existem c servidores idênticos em paralelo; a disciplina da fila é genérica, ou seja, qualquer tipo de disciplina; a capacidade do sistema é ilimitada e a população é infinita.

As filas markovianas finitas, indicadas de forma genérica por M/M/c/d, são semelhantes às infinitas, a menos da capacidade do sistema limitada a d usuários, em que $d \geq c$.

Um caso particular de filas markovianas finitas é o modelo M/M/c/c, conhecido como modelo de perda de Erlang. Ele é denominado sistema de perda por não haver espaço para formação de filas de espera, uma vez que a capacidade do sistema é igual a quantidade de servidores; logo clientes que chegam quando todos os servidores estão ocupados não são permitidos esperar pelo serviço e vão embora. A fila M/M/c/c possui inúmeras aplicações práticas, tais como no planejamento de centrais telefônicas, estacionamentos, sistemas de computação e similares.

O diagrama de transição de estados de uma fila M/M/c/c é apresentado na Figura 2.3. Neste diagrama, o estado do sistema é o número de clientes no sistema, que varia de 0 (sistema vazio) até c (capacidade máxima do sistema, que é igual ao número de servidores), λ é a taxa de chegada de clientes ao sistema, e μ é a taxa de serviço de cada servidor.

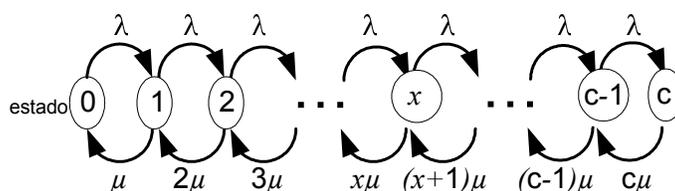


Figura 2.3: Diagrama de Transição de estados para sistema de fila M/M/c/c.

A análise de muitos sistemas de filas baseia-se principalmente no comportamento do sistema em estado de equilíbrio ou regime permanente (*steady state*), atingido após um tempo de operação suficientemente grande do sistema. Quando a fila M/M/c/c está em equilíbrio, a probabilidade de o sistema estar no estado x (independentemente de t), ou seja, a proporção de tempo que o sistema permanece no estado x é dada pela seguinte equação:

$$\pi(x) = \frac{\frac{\rho^x}{x!}}{1 + \rho + \frac{\rho^2}{2!} + \dots + \frac{\rho^c}{c!}}, \quad x = 1, 2, \dots, c, \quad \rho = \frac{\lambda}{\mu}. \quad (2.2)$$

Para saber mais sobre a teoria de filas consultar, por exemplo, Allen (1978); Asmussen (1987); Wolff (1989); Bolch et al. (1998).

Em sistemas de satélites, como o considerado neste trabalho, cada conexão pode ser representada por uma fila M/M/c/c, mas deve-se levar em conta o fato que a quantidade de tráfego em uma conexão depende da quantidade de tráfego nas outras conexões. Esta dependência será discutida no próximo capítulo.

3 MODELAGEM DO SISTEMA DE SATÉLITES

Neste capítulo descrevem-se o sistema de satélites em estudo, o modelo markoviano proposto e medidas de desempenho para análise de funcionamento deste sistema.

3.1 Sistema em estudo

Embora, a princípio, sejam necessários somente três satélites igualmente espaçados ao longo do equador, a intervalos de 120° , para cobrir toda a superfície terrestre, à exceção dos pólos (CLARKE, 1945), na prática mais satélites são usados com o objetivo de aumentar a capacidade do sistema de tratar as mensagens. Como o limite de visibilidade de cada satélite geostacionário é de $159,47^\circ$, há aproximadamente 40° de cobertura conjunta por dois satélites para um observador na linha do equador, com menos ainda em latitudes maiores. Neste sistema, a distância entre os satélites (conexão ISL) torna o tempo de transmissão de chamadas nas conexões ISL grande, comparado com sistemas com mais satélites, além de poder causar ruído nestas conexões.

Neste trabalho, considera-se uma constelação de seis satélites igualmente espaçados em uma única órbita Geoestacionária como apresentada na Figura 3.1. Por se tratar de um sistema Geoestacionário nenhuma operação de *handoff* é realizada.

Se as chegadas de chamadas ocorrem de maneira totalmente aleatória, ou seja, não são influenciadas pelos instantes de suas chegadas nem pelo tempo decorrido desde a última chegada ou pela duração desta, os intervalos entre chegadas podem ser descritos por uma distribuição exponencial. Neste caso, as chegadas seguem uma distribuição de Poisson. Analogamente, se a duração de uma chamada não é influenciada pelo intervalo de tempo em que ela ocorre nem pela duração das chamadas anteriores, ela pode ser descrita como uma distribuição exponencial.

Embora não se disponha de dados reais sobre as chegadas e a duração das chamadas, supõe-se, neste trabalho, que as chamadas chegam a cada satélite de acordo com um processo de Poisson à taxa λ e a duração das chamadas segue uma distribuição

exponencial com taxa μ . Esta suposição, não aparenta fugir muito da realidade. No entanto, uma comprovação só poderá ser obtida se dados reais estiverem disponíveis.

Admite-se que, para todos os satélites i , as quantidades de canais $C_{UDL\ i-i}$ são iguais e são denominados simplesmente C_{UDL} , a menos que, para entendimento do texto, seja necessário fazer esta distinção. Analogamente, admite-se que as quantidades de canais nas conexões ISL são iguais para quaisquer satélites $i, j, i \neq j$, e são denominados simplesmente C_{ISL} . Esta simplificação não causa perda de generalidade, e, portanto, o modelo proposto pode ser utilizado quando o número de canais UDL e/ou ISL são diferentes.

Em uma constelação de satélites de comunicação as chamadas são roteadas da sua origem ao seu destino através das conexões físicas existentes. A definição da rota por onde a chamada passará depende de vários fatores, tais como a área de menor solicitação de serviços sobre a superfície terrestre ou a capacidade física das conexões entre satélites. Neste sistema, admite-se que uma chamada entre usuários que estão nas áreas de cobertura dos satélites 1 e 4 é realizada através dos satélites 2 e 3; uma chamada entre usuários que estão nas áreas de cobertura dos satélites 2 e 5 é realizada através dos satélites 1 e 6; uma chamada entre usuários que estão nas áreas de cobertura dos satélites 3 e 6 é realizada através dos satélites 2 e 1 e que todas as outras chamadas são realizadas via a rota de menor distância entre os satélites, por exemplo, uma chamada entre usuários que estão nas áreas de cobertura dos satélites 1 e 3 é realizada através do satélite 2 (Figura 3.1).

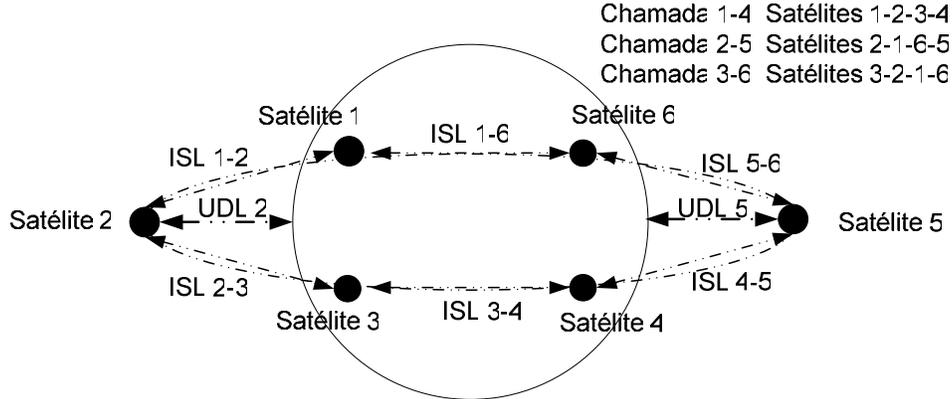


Figura 3.1 – Seis satélites em uma única órbita.

3.2 O modelo markoviano

Como no sistema em estudo supõe-se que as chamadas chegam a cada satélite de acordo com um processo de Poisson e o tempo de duração de cada chamada é exponencialmente distribuído, é possível modelá-lo como uma cadeia de Markov a tempo contínuo (CMTC). Neste trabalho utiliza-se a CMTC proposta por Zaim (2001) para modelar o sistema apresentado na Seção 3.2.

O estado “ e ” do sistema é definido pelo conjunto de valores $(n_{11}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{22}, n_{23}, n_{24}, n_{25}, n_{26}, n_{33}, n_{34}, n_{35}, n_{36}, n_{44}, n_{45}, n_{46}, n_{55}, n_{56}, n_{66})$, onde n_{ij} representa o número de chamadas bidirecionais ativas entre o satélite i e o satélite j , não importando se as chamadas originaram no satélite i ou j . Por exemplo, se $n_{14} = 1$ significa que há uma chamada utilizando um canal da conexão ISL1-4 de mão única do satélite 1 para o satélite 4. Como n_{ii} representa o número de chamadas entre dois usuários na mesma área de cobertura do satélite i , dois canais bidirecionais UDL são usados para realizar estas chamadas. Então, se, por exemplo, está ocorrendo uma chamada entre dois clientes debaixo de satélite 2, $n_{22} = 1$ e dois canais bidirecionais da conexão UDL2-2 são usados.

Logo, o espaço de estados do sistema é dado por:

$$E = \{ (n_{11}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{22}, n_{23}, n_{24}, n_{25}, n_{26}, n_{33}, n_{34}, n_{35}, n_{36}, n_{44}, n_{45}, n_{46}, n_{55}, n_{56}, n_{66}) /$$

$$n_{ij} \in \mathcal{N} \forall i, j;$$

$$2n_{11} + n_{12} + n_{13} + n_{14} + n_{15} + n_{16} \leq C_{UDL}; \quad (3.1)$$

$$n_{12} + 2n_{22} + n_{23} + n_{24} + n_{25} + n_{26} \leq C_{UDL}; \quad (3.2)$$

$$n_{13} + n_{23} + 2n_{33} + n_{34} + n_{35} + n_{36} \leq C_{UDL}; \quad (3.3)$$

$$n_{14} + n_{24} + n_{34} + 2n_{44} + n_{45} + n_{46} \leq C_{UDL}; \quad (3.4)$$

$$n_{15} + n_{25} + n_{35} + n_{45} + 2n_{55} + n_{56} \leq C_{UDL}; \quad (3.5)$$

$$n_{16} + n_{26} + n_{36} + n_{46} + n_{56} + 2n_{66} \leq C_{UDL}; \quad (3.6)$$

$$n_{12} + n_{13} + n_{14} + n_{25} + n_{26} + n_{36} \leq C_{ISL}; \quad (3.7)$$

$$n_{13} + n_{14} + n_{23} + n_{24} + n_{36} \leq C_{ISL}; \quad (3.8)$$

$$n_{14} + n_{24} + n_{34} + n_{35} \leq C_{ISL}; \quad (3.9)$$

$$n_{35} + n_{45} + n_{46} \leq C_{ISL}; \quad (3.10)$$

$$n_{15} + n_{25} + n_{46} + n_{56} \leq C_{ISL}; \quad (3.11)$$

$$n_{15} + n_{16} + n_{25} + n_{26} + n_{36} \leq C_{ISL} \}. \quad (3.12)$$

As restrições (3.1) até (3.12) são devidas ao número finito de canais nas conexões UDL e ISL e ao compartilhamento destes pelas diversas chamadas. A restrição (3.1) assegura que o número de chamadas originadas (equivalentemente, terminadas) no satélite 1 é no máximo igual à capacidade da conexão UDL daquele satélite. Note que uma chamada que se origina e termina dentro da área de abrangência do satélite 1 captura dois canais, por isso o termo $2n_{11}$ na restrição (3.1). As restrições (3.2) até (3.6) são similares a (3.1), mas correspondem aos satélites 2, 3, 4, 5 e 6, respectivamente. A restrição (3.7) assegura que o número de chamadas que usam a conexão ISL entre os satélites 1 e 2 é no máximo igual à capacidade daquela conexão. As restrições (3.8) a (3.12) são similares a (3.7), mas correspondem as conexões 2-3, 3-4, 4-5, 5-6 e 1-6, respectivamente.

A quantidade de conexões lógicas existentes é definida pela quantidade de satélites no sistema e pode ser calculada da seguinte maneira: seja k o número de satélites da constelação e N a quantidade de conexões lógicas do sistema, então $N = k(k+1)/2$.

Sejam λ_{ij} a taxa de chegada de chamadas (número médio de chamadas por unidade de tempo) e $1/\mu_{ij}$ a duração média das chamadas entre satélites i e j , respectivamente.

Quando uma conexão entre satélites recebe uma chamada, ou quando uma chamada em execução é terminada, ocorre uma transição para um novo estado. As taxas de transição entre estados, $r(e, \hat{e})$, do estado atual ($e \in E$) para o próximo estado ($\hat{e} \in E$) para esta CMTC são dadas por:

- $r(e, \hat{e}) = \lambda_{ij}, \forall i, j$, se a transição ocorre devido ao início de uma nova chamada entre os satélites i e j . Neste caso, \hat{e} é igual a e , exceto na posição correspondente ao elemento n_{ij} , que é acrescido de uma unidade;
- $r(e, \hat{e}) = n_{ij}\mu_{ij}, \forall i, j, n_{ij} > 0$, se a transição ocorre devido ao término de uma chamada entre os satélites i e j . Neste caso, \hat{e} é igual a e , exceto na posição correspondente ao elemento n_{ij} , que é decrescido de uma unidade.

A seguir, apresenta-se um exemplo que ilustra as transições entre estados e sua respectiva taxa de transição. Se o sistema está no estado $e = (0, n_{12}, 0, 0, 0, \dots, 0, 0, 0)$ com $0 < n_{12} < C_{UDL}$, com $C_{UDL} \leq C_{ISL}$, significa que estão ocorrendo n_{12} chamadas entre usuários que estão na área de cobertura dos satélites 1 e 2 e não existe nenhuma outra chamada em andamento no sistema. Se o canal ISL entre os satélites 1 e 2 ainda tem capacidade para receber mais chamadas, uma chamada será iniciada e o estado do sistema passa a ser $\hat{e} = (0, n_{12}+1, 0, 0, 0, \dots, 0, 0, 0)$ com taxa de transição λ_{12} . Analogamente, uma chamada pode terminar, e o estado do sistema passa a ser $\hat{e} = (0, n_{12}-1, 0, 0, 0, \dots, 0, 0, 0)$ com taxa de transição $n_{12}\mu_{12}$. Como não existem chamadas entre usuários que estão na área de cobertura dos outros satélites, ou seja, todos os outros canais ISL e/ou UDL entre satélites estão vazios, o único evento possível é o início de uma chamada. Por exemplo, se uma chamada é iniciada entre dois usuários que estão na área de cobertura do satélite 1, o estado do sistema passa a ser $\hat{e} = (1, n_{12}, 0, 0, 0, \dots, 0, 0, 0)$ com taxa de transição λ_{11} .

O modelo proposto pode ser facilmente estendido para constelações com mais de seis satélites.

Para fazer uma análise do comportamento do sistema a longo prazo, ou seja, quando este atingiu o equilíbrio, utilizam-se as probabilidades limites que serão apresentadas na próxima seção.

3.3 Cálculo das probabilidades estacionárias do sistema

Como apresentado na Seção 2.3, a distribuição de probabilidade estacionária, ou probabilidade limite, de uma cadeia de Markov a tempo contínuo $Y(t)$ é dada por $\pi(e) = \lim_{t \rightarrow \infty} P\{Y(t) = e\}$, em que e é o estado do sistema. Segundo Bolch et al. (1998) se a CMTC for ergódica, o sistema de equações:

$$\begin{cases} \pi Q = 0 \\ \sum_{e \in E} \pi_e = 1 \end{cases}, \quad (3.13)$$

onde a matriz Q é denominada matriz geradora infinitesimal da cadeia de Markov a tempo contínuo, tem solução única e esta solução é a probabilidade limite da cadeia. Em uma CMTC, a probabilidade limite $\pi(e)$ é a probabilidade de o sistema (em equilíbrio) estar no estado e , ou seja, a proporção do tempo que o processo permanece no estado e . Como a CMTC em estudo é ergódica, sua probabilidade limite pode ser obtida resolvendo-se (3.13).

Se o número de satélites e/ou de conexões físicas no sistema cresce, o espaço de estados aumenta, tornando intratável o sistema de equações (3.13). Uma alternativa é tratar o sistema de satélites em estudo como uma rede de filas M/M/c/c e achar as probabilidades limites utilizando uma expressão vinda da teoria de filas denominada “solução na forma de produto”. Esta expressão será apresentada na próxima seção.

3.3.1 Solução na Forma de Produto

O sistema de equações (3.13) estabelece que, em média, a taxa com a qual o sistema sai de cada estado deve ser igual à taxa com a qual o sistema entra em cada estado, ou seja,

$$\sum_{y \in E} \pi_y q_{yx} = \pi_x \sum_{y \in E} q_{xy}, \quad \forall x \in E \quad (3.14)$$

em que q_{xy} (q_{yx}) é a taxa de transição do estado x (y) para o estado y (x) e π_x (π_y) é a probabilidade limite do estado x (y), $x, y \in E$. Estas equações são chamadas equações de balanço global.

As equações de balanço global podem ser usadas para solucionar redes de filas modeladas como CMTC, ou seja, para calcular suas probabilidades limites, mas quando as redes são muito grandes, este método torna-se inviável pela quantidade de equações a serem resolvidas.

Para resolver o problema da grande quantidade de equações é necessário encontrar técnicas que diminuam o esforço computacional na resolução desses sistemas, por exemplo, usar a propriedade de balanço local.

Uma rede de filas com espaço de estados $x = (x_1, x_2, \dots, x_n)$, em que x_n é o número de clientes no nó n , tem a *propriedade de balanço local* se, para todos os nós desta rede, a taxa de transição para fora de um estado desta rede devido a uma saída (término de serviço) do nó n é igual à taxa de transição para dentro deste estado devido a uma chegada ao nó n . Assim, a taxa no qual serviços entram em um único nó da rede é igual à taxa no qual os serviços partem deste nó da rede:

$$\pi_y q_{yx} = \pi_x q_{xy}, \quad \forall x, y \in E$$

As equações de balanço local permitem uma grande simplificação em relação às equações de balanço global porque cada equação pode ser dividida em equações simples, cada uma relacionada a cada nó individual. Redes de filas que têm uma solução única das equações de balanço local são denominadas *Redes na Forma de Produto* (BOLCH ET AL., 1998).

A probabilidade limite de um estado de uma *Rede na Forma de Produto* consiste do produto das probabilidades marginais de cada nó da rede que compõe este estado. Logo, a solução na forma de produto de uma rede de filas com estado $x = (x_1, x_2, \dots, x_n)$ é dada por:

$$\pi(x) = \pi(x_1, x_2, \dots, x_n) = \frac{1}{G} [\pi(x_1) \cdot \pi(x_2) \cdot \dots \cdot \pi(x_n)], x \in E$$

em que,

$$G = \sum_{x \in E} \frac{\rho_{x_1}^{x_1}}{x_1!} \frac{\rho_{x_2}^{x_2}}{x_2!} \dots \frac{\rho_{x_n}^{x_n}}{x_n!}, \rho_i = \frac{\lambda_i}{\mu_i}, i = 1, 2, \dots, n.$$

Pode ser mostrado que um sistema de filas do tipo M/M/c com uma disciplina de fila do tipo FCFS possui a propriedade de balanço local (BOLCH ET AL., 1998),

Como a rede em estudo neste trabalho é formada por várias filas M/M/c/c, para que a solução na forma de produto seja utilizada para o cálculo das probabilidades limites desta rede é necessário utilizar a propriedade de reversibilidade.

Um processo estocástico $X(t)$ é reversível se $[X(t_1), X(t_2), \dots, X(t_n)]$ possui a mesma distribuição conjunta $[X(\tau - t_1), X(\tau - t_2), \dots, X(\tau - t_n)]$ para todo $t_1, t_2, \dots, t_n, \tau \in T$. Um processo de Markov estacionário é reversível se e somente se existe um conjunto de números positivos $\pi(j), j \in E$, cuja soma é unitária e satisfaz as condições de balanço local $[\pi(j)q(j,k) = \pi(k)q(k,j)], j, k \in E$, em que $\pi(j)$ é a distribuição estacionária do

processo. Então se uma cadeia de Markov tem balanço local, ela é reversível (KELLY, 1979).

Se o sistema é reversível, têm-se os seguintes teoremas apresentados em Wolff (1989).

Teorema 1: Se $\{X_1(t)\}$ e $\{X_2(t)\}$ são processos estocásticos independentes sendo cada um reversível, o vetor de valores do processo $\{X(t)\}$, definido por $X(t) = \{X_1(t), X_2(t)\}$ para todo t , também será reversível. Assim, se dois processos estocásticos independentes são cada um reversíveis, então o processo estocástico resultante da união desses dois processos também será reversível.

Teorema 2: Seja uma cadeia de Markov reversível com distribuição estacionária π_x , trunca-se o seu espaço de estados e cria-se uma nova cadeia com espaço de estados E , para qualquer E desde que a nova cadeia seja irredutível. A nova cadeia é reversível e

tem uma distribuição estacionária dada por $\frac{\pi_x}{\sum_{y \in E} \pi_y}$. Ou seja, se de uma cadeia de

Markov reversível, com distribuição estacionária, for criada uma nova cadeia de Markov, através do truncamento do espaço de estados da cadeia original, a nova cadeia de Markov será reversível e terá a mesma distribuição estacionária, respectiva a seus elementos, desde que a nova cadeia seja irredutível.

Considera-se uma rede de filas em que cada fila é do tipo M/M/c, ou seja, as chegadas e os termos de serviços seguem um processo de Poisson e a capacidade de cada fila é irrestrita. Pelo Teorema 1, conclui-se que, como a fila M/M/c é reversível, a rede de filas resultante da união de filas M/M/c também é reversível e, portanto, tem a propriedade de balanço local. Se truncarmos o espaço de estados desta rede de filas, limitando o número de clientes (chamadas) ao número de servidores (canais), pelo Teorema 2, a nova rede de filas, em que cada fila será do tipo M/M/c/c, continuará reversível e terá a propriedade de balanço local, sendo possível utilizar a solução na forma de produto para resolver essa nova CMTC.

Essa idéia foi utilizada no cálculo das probabilidades limites do sistema de satélites modelado neste trabalho. As probabilidades limites foram utilizadas nos cálculos de medidas de desempenho apresentadas na próxima seção.

3.4 Medidas de desempenho

Nesta seção, descrevem-se as expressões para obtenção das seguintes medidas de desempenho do funcionamento do sistema em estudo: o número médio de chamadas entre os satélites i e j , o número médio de canais usados em cada conexão, a probabilidade de bloqueio e a taxa de rejeição de chamadas no sistema e em cada conexão, e a taxa efetiva de chegada de chamadas em cada conexão.

O número médio de chamadas entre os satélites i e j é dado pelo somatório, sob todos os estados do sistema, do produto entre o número de chamadas ativas entre estes satélites em cada estado e , $n_{ij}(e)$, e sua respectiva probabilidade limite π_e , ou seja:

$$\bar{X}(n_{ij}) = \sum_{\forall e \in E} n_{ij}(e) \pi_e, \quad 1 \leq i \leq j \leq 6.$$

O número médio de canais utilizados em cada conexão física é dado pelo somatório do número médio de chamadas entre satélites que usam estes canais, sendo que chamadas que ocorrem entre dois usuários na mesma área de cobertura de um determinado satélite usam dois canais UDL. Então, tem-se que:

$$\overline{N_{UDL1}} = 2\bar{X}(n_{11}) + \bar{X}(n_{12}) + \bar{X}(n_{13}) + \bar{X}(n_{14}) + \bar{X}(n_{15}) + \bar{X}(n_{16})$$

$$\overline{N_{UDL2}} = 2\bar{X}(n_{22}) + \bar{X}(n_{12}) + \bar{X}(n_{23}) + \bar{X}(n_{24}) + \bar{X}(n_{25}) + \bar{X}(n_{26})$$

$$\overline{N_{UDL3}} = 2\bar{X}(n_{33}) + \bar{X}(n_{13}) + \bar{X}(n_{23}) + \bar{X}(n_{34}) + \bar{X}(n_{35}) + \bar{X}(n_{36})$$

$$\overline{N_{UDL4}} = 2\bar{X}(n_{44}) + \bar{X}(n_{14}) + \bar{X}(n_{24}) + \bar{X}(n_{34}) + \bar{X}(n_{45}) + \bar{X}(n_{46})$$

$$\overline{N_{UDL5}} = 2\bar{X}(n_{55}) + \bar{X}(n_{15}) + \bar{X}(n_{25}) + \bar{X}(n_{35}) + \bar{X}(n_{45}) + \bar{X}(n_{56})$$

$$\overline{N_{UDL6}} = 2\bar{X}(n_{66}) + \bar{X}(n_{16}) + \bar{X}(n_{26}) + \bar{X}(n_{36}) + \bar{X}(n_{46}) + \bar{X}(n_{56})$$

$$\begin{aligned}
\overline{N_{ISL\ 1-2}} &= \bar{X}(n_{12}) + \bar{X}(n_{13}) + \bar{X}(n_{14}) + \bar{X}(n_{25}) + \bar{X}(n_{26}) + \bar{X}(n_{36}) \\
\overline{N_{ISL\ 2-3}} &= \bar{X}(n_{13}) + \bar{X}(n_{14}) + \bar{X}(n_{23}) + \bar{X}(n_{24}) + \bar{X}(n_{36}) \\
\overline{N_{ISL\ 3-4}} &= \bar{X}(n_{14}) + \bar{X}(n_{24}) + \bar{X}(n_{34}) + \bar{X}(n_{35}) \\
\overline{N_{ISL\ 4-5}} &= \bar{X}(n_{35}) + \bar{X}(n_{45}) + \bar{X}(n_{46}) \\
\overline{N_{ISL\ 5-6}} &= \bar{X}(n_{15}) + \bar{X}(n_{25}) + \bar{X}(n_{46}) + \bar{X}(n_{56}) \\
\overline{N_{ISL\ 1-6}} &= \bar{X}(n_{15}) + \bar{X}(n_{16}) + \bar{X}(n_{25}) + \bar{X}(n_{26}) + \bar{X}(n_{36})
\end{aligned}$$

A probabilidade de bloqueio de chamadas na conexão i - j , $\pi_b(i, j)$ é calculada somando-se todas as probabilidades limites dos estados em que a conexão i - j deve ser bloqueada por falta de recursos (canais) disponíveis para atenderem chamadas. Seja $B_{ij} = \{e | e \in E \text{ e } e + 1_{ij} \notin E\}$, onde 1_{ij} representa um vetor com zero para todas as variáveis aleatórias exceto a variável aleatória n_{ij} , que é 1, então a probabilidade de bloqueio na conexão i - j é dada por:

$$\pi_b(i, j) = \sum_{\forall e \in B_{ij}} \pi_e. \quad (3.15)$$

A taxa de rejeição de chamadas na conexão i - j é dada pelo produto entre a taxa de chegadas de chamadas e a probabilidade de bloqueio de chamadas nesta conexão. Então, tem-se que:

$$\theta_{ij} = \lambda_{ij} \pi_b(i, j),$$

em que λ_{ij} é a taxa de chegada de chamadas na conexão i - j .

Logo, a taxa total de rejeição do sistema é dada pelo somatório das taxas de rejeição de chamadas em todas as conexões do sistema, ou seja,

$$\theta = \sum_{i,j} \theta_{ij} = \sum_{i,j} \lambda_{ij} \pi_b(i,j).$$

Definindo a taxa total de chegadas no sistema como o somatório das taxas de chegada de chamadas em todas as conexões do sistema, ou seja,

$$\lambda = \sum_{i,j} \lambda_{ij},$$

tem-se que a probabilidade de bloqueio do sistema é dada pela razão entre a taxa de rejeição de chamadas e a taxa de chegada de chamadas do sistema,

$$\pi_b = \frac{\sum_{i,j} \lambda_{ij} \pi_b(i,j)}{\lambda}.$$

A taxa efetiva de chegada de chamadas (taxa de atendimento) na conexão $i-j$ é dada pela diferença entre a taxa de chegada e a taxa de rejeição de chamadas naquela conexão, ou seja,

$$\lambda'_{ij} = \lambda_{ij}(1 - \pi_b(i,j)) = \lambda_{ij} - \lambda_{ij} \pi_b(i,j) = \lambda_{ij} - \theta_{ij}.$$

4 SISTEMA DE SATÉLITE COM *HANDOFF*

Neste capítulo, considera-se que na constelação de seis satélites, apresentada no capítulo 3, os satélites não do tipo LEO. Estes satélites estão localizados em órbitas entre 500 e 2000 km de altitude. Como estas órbitas estão próximas à Terra, estes satélites têm velocidade angular maior do que a velocidade de rotação da Terra, pois, caso contrário, a força da gravidade os puxaria para o solo, destruindo-os. Neste sistema, operações de *handoff* são realizadas.

Na próxima seção descreve-se o sistema em estudo e na seção 4.2 a modelagem markoviana proposta para este sistema.

4.1 Sistema em estudo

Considere uma constelação de seis satélites como mostrado na Figura 4.1. Nesta constelação, a área de cobertura de cada satélite intercepta a área de cobertura dos satélites adjacentes numa região de interferência de sinal denominada área de transferência de chamadas. Um cliente que esteja situado nesta área está conectado a apenas um satélite, cujo sinal seja de melhor qualidade. À medida que os satélites passam sobre a superfície terrestre, uma chamada que está acontecendo na área de cobertura de um satélite passará para área de cobertura do próximo satélite.

Denomina-se *handoff* à transferência de chamadas que ocorre de um satélite para outro satélite, devido à transferência do sinal da área de cobertura de um satélite para a área de cobertura do próximo satélite. O *handoff* acontece quando um usuário passa para fora do alcance de uma área de cobertura e entra no alcance da área de cobertura do satélite seguinte, logo o sinal passa de uma área de cobertura para a próxima. O *handoff* deve ser imperceptível aos usuários e não gerar perda de serviço, a menos que o usuário mova-se para fora do alcance de todas as áreas de cobertura.

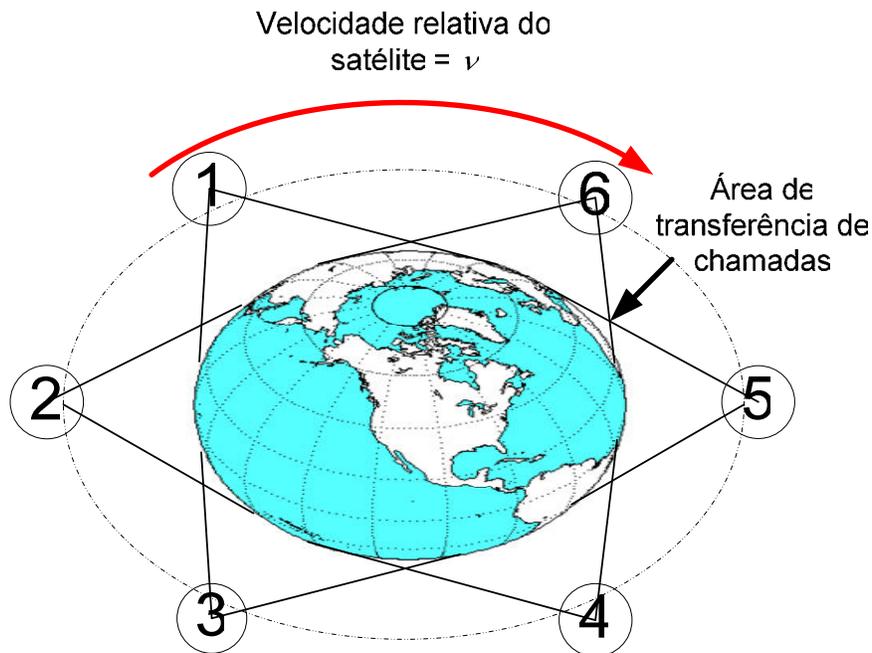


Figura 4.1 – Sistema de seis satélites em órbita baixa

Na modelagem proposta, considera-se:

- A taxa de chegada λ de novas chamadas em cada satélite permanece constante à medida que os satélites movem-se em torno da Terra, ou seja, a taxa de chegada de chamadas entre os satélites i e j é dada por $\lambda_{ij} = \lambda f_{ij}$, onde f_{ij} é a probabilidade de que uma chamada originada na área de cobertura do satélite i seja para um cliente na área de cobertura do satélite j .
- Os clientes ativos na área de cobertura do satélite i são considerados uniformemente distribuídos na área de cobertura deste satélite. Logo, a taxa de *handoff* do satélite i para o seguinte é constante, independente do satélite sob o qual o usuário está posicionado e proporcional ao número de chamadas existentes na área de cobertura do satélite i . Esta aproximação será adequada quando as taxas de recebimento de chamadas for bem maior que as de *handoff*, ou seja, quando a fração das chamadas sujeitas à *handoff* for pequena.

- Todos os satélites são idênticos e estão uniformemente distribuídos numa mesma órbita. Segundo Henderson e Katz (2000), na maioria das constelações comerciais de satélite LEO, os satélites são uniformemente distribuídos em torno da órbita.

4.2 Modelagem markoviana

Sejam A a área de cobertura de um satélite, v_A a velocidade desta área em relação ao solo, Δt um intervalo de tempo em que o satélite moveu sobre a superfície terrestre, ΔL a distância que a área de cobertura do satélite se move no intervalo de tempo Δt , e ΔA a parcela da área de cobertura do satélite que foi transferida para o próximo satélite. O movimento do satélite está ilustrado na Figura 4.2.

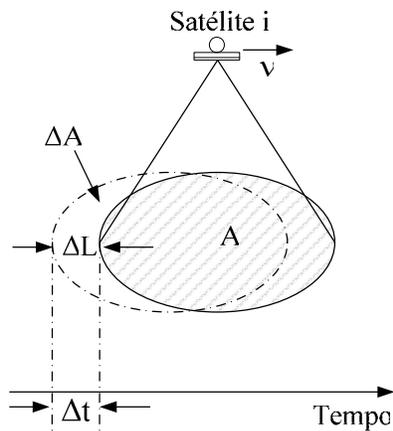


Figura 4.2 – Itens da modelagem do *handoff*.

Fonte: Adaptada de Zaim (2001).

Clientes localizados na parcela ΔA da área de cobertura original de um satélite têm suas chamadas transferidas para o próximo satélite. Considerando-se que estes clientes são uniformemente distribuídos, a probabilidade g que uma chamada seja transferida para o próximo satélite dentro do intervalo de tempo Δt é dada por:

$$g = \frac{\Delta A}{A} \quad (4.1)$$

De uma forma geral, para pequenos intervalos de tempo, a variação da área ΔA é proporcional à área A e ao deslocamento da área de cobertura do satélite ΔL ; $\Delta A = \beta A \Delta L$, em que β depende da forma da área de cobertura do satélite (se esta área for retangular, $\beta=1$). Tem-se que $\Delta L = v \Delta t$. Então:

$$g = \beta \cdot v_A \cdot \Delta t \quad (4.2)$$

A taxa de *handoff*, denominada α , é dada por:

$$\alpha = \lim_{\Delta t \rightarrow 0} \frac{g}{\Delta t} = \beta v_A \quad (4.3)$$

Como, neste trabalho, considera-se um modelo a tempo contínuo, chamadas entre usuários na área de cobertura do satélite i e usuários na área de cobertura do satélite j serão transferidas, devido ao *handoff*, se pelo menos um usuário de um dos satélites estiver na área de transferência de chamadas. Neste caso, a probabilidade de que uma chamada seja transferida, quando o cliente debaixo do satélite i estiver na área de transferência de chamadas e o cliente debaixo do satélite j não estiver na área de transferência de chamadas, é $g(1-g) = g - g^2$. Se fosse permitido que os usuários dos dois satélites estivessem nas suas respectivas áreas de transferência de chamadas ao mesmo tempo, a probabilidade de transferência de chamadas seria g^2 que da Equação 4.3 é $o(\Delta t)$. Por isso *handoffs* simultâneos não são considerados. Então, a probabilidade de que exatamente uma chamada ocorrendo entre os satélites i e j seja transferida para um satélite adjacente é dada por $2g(1-g) = 2g - 2g^2 = 2g$.

Quando existirem n usuários ativos num dado satélite, cada qual executando uma chamada, a taxa de transferência de qualquer um desses usuários para o próximo satélite é dada por αn .

A constelação de seis satélites apresentada na Figura 4.1 pode ser modelada como uma cadeia de Markov a tempo contínuo com o mesmo espaço de estados E e as mesmas

taxas de transição $r(e, \hat{e})$ do Capítulo 3, às quais devem ser adicionadas taxas de transição devido ao *handoff*.

As taxas de transição $r(e, e_h)$, devido ao *handoff*, do estado atual $e \in E$ para o próximo estado $e_h \in E$ é dada por:

- $r(e, e_h) = \alpha n_{ij}$, quando as chamadas são transferidas, devido ao *handoff*, do satélite i para o satélite j que vem logo em seguida. Neste caso e é igual à e_h , exceto na posição que corresponde ao elemento n_{ij} , que é decrescido de uma unidade, devido a chamada que foi transferida, e na posição do satélite que recebeu a chamada, que é aumentada de uma unidade.
- $r(e, e_h) = 2\alpha n_{ii}$, quando ambos os usuários são servidos pelo mesmo satélite. Neste caso e é igual à e_h , exceto na posição correspondente ao elemento n_{ii} , que é decrescida de uma unidade, e da posição do satélite que recebeu a chamada transferida, que é aumentada de uma unidade.

Para cada estado $e = (n_{11}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{22}, n_{23}, n_{24}, n_{25}, n_{26}, n_{33}, n_{34}, n_{35}, n_{36}, n_{44}, n_{45}, n_{46}, n_{55}, n_{56}, n_{66}) \in E$, na Tabela 4.1 apresentam-se as transições devido ao *handoff*. Nesta tabela, o próximo estado e_h está escrito em função do estado atual e e do vetor $1_{i,j}$, que possui a mesma dimensão do estado do sistema, com todas as componentes iguais a zero, exceto na posição $i-j$ que é igual a 1. Cada linha da tabela mostra as condições que o estado atual deve satisfazer para que ocorra a transição, o próximo estado e a taxa da transição.

Tabela 4.1 - Transições devido ao *Handoff*

Condição	Próximo estado (e_h)	Taxa
$j \leq 6, i < j, n_{ij} > 0, e - 1_{ij} + 1_{i+1,j} \in E$	$e - 1_{ij} + 1_{i+1,j}$	αn_{ij}
$j < 6, i < j, n_{ij} > 0, e - 1_{ij} + 1_{ij+1} \in E$	$e - 1_{ij} + 1_{ij+1}$	αn_{ij}
$j = 6, i < j, n_{ij} > 0, e - 1_{ij} + 1_{i,1} \in E$	$e - 1_{ij} + 1_{i,1}$	αn_{ij}
$i < 6, j = i, n_{ii} > 0, e - 1_{i,i} + 1_{i,i+1} \in E$	$e - 1_{i,i} + 1_{i,i+1}$	$2 \alpha n_{ii}$
$i = 6, j = 6, n_{ij} > 0, e - 1_{6,6} + 1_{1,6} \in E$	$e - 1_{6,6} + 1_{1,6}$	$2 \alpha n_{ii}$

Para todo $i, j, 1 \leq i \leq j \leq 6$, a taxa total de chegada de chamadas λ_{ij}'' entre os satélites i e j será a soma da taxa de chegada de novas chamadas λ_{ij} e da taxa de chegada de chamadas devido ao *handoff* αn_{ij} , se as chamadas chegam de uma conexão ISL, ou $2\alpha n_{ii}$, se as chamadas chegam de uma conexão UDL. Similarmente, a taxa total de duração de chamadas $n_{ij} \mu_{ij}''$ entre os satélites i e j , quando existirem n_{ij} chamadas na área de cobertura do satélite i , será dada pela soma da taxa de duração das novas chamadas $n_{ij} \mu_{ij}$ e da taxa de duração das chamadas devidas ao *handoff* $2\alpha n_{ij}$, isto porque todas as chamadas podem ser transferidas para duas conexões. As chamadas que ocorrem em uma conexão $(i, j), i \neq j$, podem ser transferidas para as conexões: $(i+1, j)$ ou $(i, j+1)$, com exceção das chamadas na conexão $(1, 6)$ que podem ser transferidas para as conexões $(1, 1)$ ou $(2, 6)$. As chamadas que ocorrem em uma conexão $(i, j), i = j$, podem ser transferidas para a conexão $(i, i+1)$ ou $(i+1, i)$, mas, neste trabalho, n_{ij} representa o número de chamadas bidirecionais entre os satélites i e j , não importando se elas se originaram no satélite i ou j , ou seja, considera-se $(i, i+1) = (i+1, i)$. As chamadas na conexão $(6, 6)$ só podem ser transferidas para a conexão $(1, 6)$.

O cálculo das probabilidades estacionárias do sistema é idêntico ao apresentado na Seção 3.4 exceto que λ_{ij} deve ser substituído por λ_{ij}'' e $n_{ij} \mu_{ij}$ deve ser substituído por $n_{ij} \mu_{ij}''$.

5 MÉTODO DE DECOMPOSIÇÃO

A dificuldade de se representar sistemas de satélites utilizando o modelo markoviano apresentado no Capítulo 3 é que seu espaço de estados cresce muito quando o número de satélites e/ou de canais cresce. Para resolver este problema, técnicas de agregação e desagregação, que permitam o tratamento de modelos markovianos com um grande número de estados, têm sido propostas por vários autores, tais como: Stewart e Wu (1996); Kafety et al. (1996); Strelen (1997); Strelen et al. (1997); Zaim(2001); Marek e Pultarova (2005).

Neste trabalho utiliza-se o método de decomposição proposto por Zaim (2001). Nesta referência, o autor decompõe um sistema de seis satélites em dois subsistemas de três satélites, e estes são modelados por cadeias de Markov a tempo contínuo. Um algoritmo iterativo para calcular as probabilidades de bloqueio em cada conexão dos subsistemas é proposto. Em 2003, Zaim et al. apresentam uma nova expressão para o cálculo das probabilidades de bloqueio dos satélites que estão em subsistemas diferentes.

Na Seção 5.1 deste capítulo, decompõe-se a constelação de satélites em estudo em dois subsistemas menores conforme proposto em Zaim (2001). Zaim (2001) e Zaim et al. (2003) não apresentam expressões para o cálculo da taxas de duração de chamadas em cada subsistema. Na Seção 5.2, propõem-se expressões para obtenção destas taxas. Na Seção 5.3 apresenta-se o algoritmo de decomposição desenvolvido por Zaim (2001) para obtenção das probabilidades de bloqueio do sistema decomposto, e propõem-se novas expressões para o cálculo destas probabilidades. Na Seção 5.4 e 5.5 são descritas, respectivamente, as equações, propostas por Zaim (2001) e Zaim et al. (2003), para o cálculo das probabilidades de bloqueio entre satélites que estão em subsistemas diferentes.

5.1 Método de decomposição

No método proposto por Zaim (2001) decompõe-se a constelação de satélites em subsistemas menores, com três satélites se o número de satélites na constelação for

múltiplo de três ou, caso contrário, com dois satélites, e utiliza-se um algoritmo para obtenção das probabilidades de bloqueio de chamadas deste sistema decomposto.

Neste trabalho, decompõe-se a constelação de seis satélites em estudo em dois subsistemas com três satélites cada, o subsistema 1 composto pelos satélites 1, 2 e 3 e o subsistema 2 composto pelos satélites 4, 5 e 6, conforme apresentado na Figura 5.1.

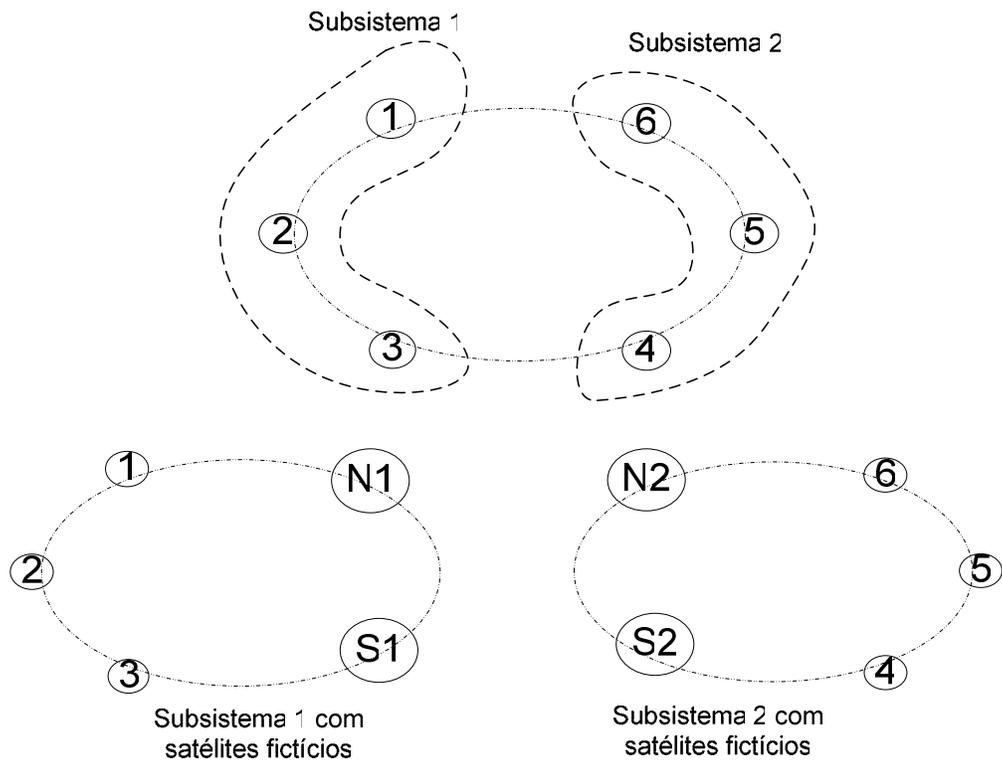


Figura 5.1 – Sistema de seis satélites decomposto

Ao subsistema 1 são adicionados os satélites fictícios N1 e S1. Desta forma, considerando-se isoladamente este subsistema, tem-se que chamadas de/para fora deste subsistema, ou seja, do satélite i , $i \in \{1, 2, 3\}$, para o subsistema 2, ou vice-versa, são realizadas através dos satélites fictícios N1 ou S1. Considerando-se o mesmo roteamento apresentado no Capítulo 3, admite-se que, através do satélite fictício N1, passam as chamadas entre os satélites 1 e 5, 1 e 6, 2 e 5, 2 e 6, e 3 e 6. E, através do satélite fictício S1, passam as chamadas entre os satélites 1 e 4, 2 e 4, 3 e 4, e 3 e 5.

Analogamente, ao subsistema 2 foram adicionados os satélites fictícios N2 e S2. Desta forma, considerando-se isoladamente este subsistema, tem-se que chamadas de/para fora deste subsistema, ou seja, do satélite $j, j \in \{4, 5, 6\}$, para o subsistema 1, ou vice-versa, são realizadas através dos satélites fictícios N2 ou S2. Admite-se que, através do satélite fictício N2, passam as chamadas entre os satélites 5 e 1, 5 e 2, 6 e 1, 6 e 2, e 6 e 3. E, através do satélite fictício S2, passam as chamadas entre os satélites 4 e 1, 4 e 2, 4 e 3, e 5 e 3. Admite-se, também, que, através do satélite N2, não passam chamadas de/para o satélite 4, e que, através do satélite S2, não passam chamadas de/para o satélite 6.

Como os satélites N1, N2, S1 e S2 são fictícios, admite-se que não ocorram chamadas entre eles e a Terra e, desta forma, não existem canais UDL fazendo estas conexões.

O espaço de estados do subsistema 1 é dado por:

$$E = \{(n_{11}, n_{12}, n_{13}, n_{1N1}, n_{1S1}, n_{22}, n_{23}, n_{2N1}, n_{2S1}, n_{33}, n_{3N1}, n_{3S1}) \mid n_{ij} \in \mathcal{N}^* \forall i, j;$$

$$2n_{11} + n_{12} + n_{13} + n_{1N1} + n_{1S1} \leq C_{UDL} \quad (5.1)$$

$$n_{12} + 2n_{22} + n_{23} + n_{2N1} + n_{2S1} \leq C_{UDL} \quad (5.2)$$

$$n_{13} + n_{23} + 2n_{33} + n_{3N1} + n_{3S1} \leq C_{UDL} \quad (5.3)$$

$$n_{1N1} + n_{2N1} + n_{3N1} \leq C_{ISL} \quad (5.4)$$

$$n_{12} + n_{13} + n_{1S1} + n_{2N1} + n_{3N1} \leq C_{ISL} \quad (5.5)$$

$$n_{13} + n_{23} + n_{1S1} + n_{2S1} + n_{3N1} \leq C_{ISL} \quad (5.6)$$

$$n_{1S1} + n_{2S1} + n_{3S1} \leq C_{ISL} \}. \quad (5.7)$$

As restrições (5.1) a (5.7) são análogas às restrições (3.1) a (3.12), apresentadas no Capítulo 3 para o sistema sem decomposição, e devem-se ao compartilhamento das conexões UDL e ISL. A restrição (5.1) assegura que o número de chamadas originadas (equivalentemente, terminadas) no satélite 1 é no máximo igual à capacidade da conexão UDL daquele satélite. As restrições (5.2) e (5.3) são similares a (5.1), mas correspondem aos satélites 2 e 3, respectivamente. As restrições (5.4) a (5.7) asseguram que o número de chamadas que usam a conexão ISL, entre dois satélites do subsistema

1, é no máximo igual à capacidade daquela conexão. A restrição (5.4) refere-se ao compartilhamento da conexão entre os satélites 1 e N1; por esta conexão passam as chamadas entre os satélites 1 e N1, entre os satélites 2 e N1, e entre os satélites 3 e N1. As restrições (5.5), (5.6) e (5.7) referem-se ao compartilhamento das conexões entre os satélites 1 e 2, 2 e 3 e 3 e S1, respectivamente.

As novas taxas de chegada de chamadas, $\lambda_{iz'}$, $z' \in \{N1, S1\}$, do subsistema 1 são dadas por:

$$\lambda_{1N1} = (1 - p'_{16}) \lambda_{16} + (1 - p'_{15}) \lambda_{15} \quad (5.8)$$

$$\lambda_{1S1} = (1 - p'_{14}) \lambda_{14} \quad (5.9)$$

$$\lambda_{2N1} = (1 - p'_{26}) \lambda_{26} + (1 - p'_{25}) \lambda_{25} \quad (5.10)$$

$$\lambda_{2S1} = (1 - p'_{24}) \lambda_{24} \quad (5.11)$$

$$\lambda_{3N1} = (1 - p'_{36}) \lambda_{36} \quad (5.12)$$

$$\lambda_{3S1} = (1 - p'_{34}) \lambda_{34} + (1 - p'_{35}) \lambda_{35} \quad (5.13)$$

Nas equações (5.8) a (5.13), a variável p'_{ij} , $i \in \{1, 2, 3\}$, $j \in \{4, 5, 6\}$, representa a probabilidade de bloqueio de chamadas entre o satélite i no subsistema 1 e o satélite j no subsistema 2 devido à falta de recursos, ou seja, canais disponíveis para que as chamadas sejam realizadas, nas conexões do subsistema 2. Através da equação (5.8) obtém-se a taxa de chegada de chamadas entre os satélites 1 e N1. Como citado anteriormente, pelo satélite N1 passam as ligações de/para o satélite 1 para os/dos satélites 5 e 6, então, uma chamada só ocorrerá na conexão entre os satélites 1 e N1 se não for bloqueada entre os satélites 1 e 6 ou entre os satélites 1 e 5. As equações (5.9) a (5.13) foram obtidas de maneira análoga.

O espaço de estados do subsistema 2 é dado por:

$$E = \{(n_{44}, n_{45}, n_{46}, n_{4N2}, n_{4S2}, n_{55}, n_{56}, n_{5N2}, n_{5S2}, n_{66}, n_{6N2}, n_{6S2}) \mid n_{ij} \in \mathcal{N} \forall i, j; \\ 2n_{44} + n_{45} + n_{46} + n_{4N2} + n_{4S2} \leq C_{UDL} \quad (5.14)$$

$$n_{45} + 2n_{55} + n_{56} + n_{5N2} + n_{5S2} \leq C_{UDL} \quad (5.15)$$

$$n_{46} + n_{56} + 2n_{66} + n_{6N2} + n_{6S2} \leq C_{UDL} \quad (5.16)$$

$$n_{4S2} + n_{5S2} \leq C_{ISL} \quad (5.17)$$

$$n_{45} + n_{46} + n_{5S2} \leq C_{ISL} \quad (5.18)$$

$$n_{46} + n_{56} + n_{5N2} \leq C_{ISL} \quad (5.19)$$

$$n_{6N2} + n_{5N2} \leq C_{ISL} \quad (5.20)$$

As restrições (5.17) até (5.20) são análogas às restrições (5.1) a (5.7) e, portanto, têm as mesmas interpretações.

As novas taxas de chegada de chamadas de/para o subsistema 2, $\lambda_{iz''}, z'' \in \{N2, S2\}$, são dadas por:

$$\lambda_{4N2} = 0 \quad (5.21)$$

$$\lambda_{4S2} = (1 - p''_{14}) \lambda_{14} + (1 - p''_{24}) \lambda_{24} + (1 - p''_{34}) \lambda_{34} \quad (5.22)$$

$$\lambda_{5N2} = (1 - p''_{15}) \lambda_{15} + (1 - p''_{25}) \lambda_{25} \quad (5.23)$$

$$\lambda_{5S2} = (1 - p''_{35}) \lambda_{35} \quad (5.24)$$

$$\lambda_{6N2} = (1 - p''_{16}) \lambda_{16} + (1 - p''_{26}) \lambda_{26} + (1 - p''_{36}) \lambda_{36} \quad (5.25)$$

$$\lambda_{6S2} = 0 \quad (5.26)$$

As equações (5.21) a (5.26) são análogas às equações (5.8) a (5.13) e, têm, portanto, as mesmas interpretações. Nestas equações, $p''_{ij}, i \in \{1, 2, 3\}, j \in \{4, 5, 6\}$, representa a probabilidade de bloqueio de chamadas entre o satélite i no subsistema 1 e o satélite j no subsistema 2 devido à falta de recursos no subsistema 1. Como citado anteriormente, admite-se que através do satélite N2 não passam chamadas de/para o satélite 4 e através do satélite S2 não passam chamadas de/para o satélite 6, desta forma, $\lambda_{4N2} = \lambda_{6S2} = 0$ (equações (5.21) e (5.26)).

Zaim (2001) não apresenta uma formulação para obtenção da duração média das chamadas entre os satélites de cada sistema e os satélites fictícios. Na próxima seção, propõem-se expressões para o cálculo destas taxas. Estas expressões são necessárias para o cálculo das probabilidades limites dos subsistemas 1 e 2.

5.2 Duração média das chamadas

Considerando-se o mesmo roteamento de chamadas apresentado na Seção 5.2, tem-se as seguintes equações para o cálculo das taxas de duração de chamadas dos subsistemas 1 e 2.

Taxas de duração de chamadas $\mu_{iz'}$, $i \in \{1, 2, 3\}$ e $z' \in \{N1, S1\}$, do subsistema 1:

$$\mu_{1N1} = [(1 - p'_{16}) \lambda_{16} \mu_{16} + (1 - p'_{15}) \lambda_{15} \mu_{15}] / \lambda_{1N1} \quad (5.27)$$

$$\mu_{1S1} = \mu_{14} \quad (5.28)$$

$$\mu_{2N1} = [(1 - p'_{26}) \lambda_{26} \mu_{26} + (1 - p'_{25}) \lambda_{25} \mu_{25}] / \lambda_{2N1} \quad (5.29)$$

$$\mu_{2S1} = \mu_{24} \quad (5.30)$$

$$\mu_{3N1} = \mu_{36} \quad (5.31)$$

$$\mu_{3S1} = [(1 - p'_{34}) \lambda_{34} \mu_{34} + (1 - p'_{35}) \lambda_{35} \mu_{35}] / \lambda_{3S1} \quad (5.32)$$

Taxas de duração de chamadas $\mu_{jz''}$, $j \in \{4, 5, 6\}$ e $z'' \in \{N2, S2\}$, do subsistema 2:

$$\mu_{4N2} = 0 \quad (5.33)$$

$$\mu_{4S2} = [(1 - p''_{14}) \lambda_{14} \mu_{14} + (1 - p''_{24}) \lambda_{24} \mu_{24} + (1 - p''_{34}) \lambda_{34} \mu_{34}] / \lambda_{4S2} \quad (5.34)$$

$$\mu_{5N2} = [(1 - p''_{15}) \lambda_{15} \mu_{15} + (1 - p''_{25}) \lambda_{25} \mu_{25}] / \lambda_{5N2} \quad (5.35)$$

$$\mu_{5S2} = \mu_{35} \quad (5.36)$$

$$\mu_{N26} = [(1 - p''_{16}) \lambda_{16} \mu_{16} + (1 - p''_{26}) \lambda_{26} \mu_{26} + (1 - p''_{36}) \lambda_{36} \mu_{36}] / \lambda_{N26} \quad (5.37)$$

$$\mu_{S26} = 0 \quad (5.38)$$

Nas equações (5.27) a (5.38) o termo $[(1 - p''_{ij}) \lambda_{ij}] / \lambda_{iz}$, i e j , $1 \leq i \leq 3 < j \leq 6$, representa a probabilidade de chegada de chamadas entre os satélites i e j , que pertencem a

subsistemas diferentes, e z , $z \in \{N1, S1, N2, S2\}$, o satélite fictício, através do qual passam as chamadas de/para fora do subsistema considerado para fazer as ligações entre i e j . Assim, $[(1 - p'_{ij}) \lambda_{ij} \mu_{ij}] / \lambda_{iz}$ representa a taxa efetiva de duração de chamada entre os satélites i e j .

5.3 Algoritmo de decomposição

Nesta seção, descreve-se o algoritmo de decomposição desenvolvido por Zaim (2001) para obtenção das probabilidades de bloqueio de chamadas nas conexões do sistema decomposto.

As expressões para o cálculo das probabilidades de bloqueio entre satélites que estão em subsistemas diferentes, apresentadas junto com o algoritmo, são contribuições do presente trabalho.

No passo inicial do algoritmo, faz-se $p_{ij} = p'_{ij} = p''_{ij} = 0$ para todo i e j , $1 \leq i \leq 3 \leq j \leq 6$; $p_{iz} = 0$, $i \in \{1, 2, 3\}$, $z' \in \{N1, S1\}$; $p_{iz'} = 0$, $i \in \{4, 5, 6\}$, $z'' \in \{N2, S2\}$.

Na primeira iteração, obtêm-se os valores das taxas de chegadas de chamada (equações (5.8) a (5.13)) e das taxas de duração de chamadas (equações (5.27) a (5.32)) do subsistema 1. Resolve-se, então, o subsistema 1, ou seja, obtêm-se as probabilidades limites dos estados do subsistema 1. A seguir, calculam-se as probabilidades de bloqueio de chamadas nas conexões do subsistema 1, através da expressão (3.15) apresentada na Seção 3.5 do Capítulo 3, e as probabilidade de bloqueio de chamadas, p_{ij} , $1 \leq i \leq 3 < j \leq 6$, nas conexões entre os subsistemas (expressões 5.39 a 5.47).

$$p_{14} = 1 - (1 - p_{1S1}) (1 - p_{4S2}) \quad (5.39)$$

$$p_{15} = 1 - (1 - p_{1N1}) (1 - p_{5N2}) \quad (5.40)$$

$$p_{16} = 1 - (1 - p_{1N1}) (1 - p_{6N2}) \quad (5.41)$$

$$p_{24} = 1 - (1 - p_{2S1}) (1 - p_{4S2}) \quad (5.42)$$

$$p_{25} = 1 - (1 - p_{2N1}) (1 - p_{5N2}) \quad (5.43)$$

$$p_{26} = 1 - (1 - p_{2N1}) (1 - p_{6N2}) \quad (5.44)$$

$$p_{34} = 1 - (1 - p_{3S1}) (1 - p_{4S2}) \quad (5.45)$$

$$p_{35} = 1 - (1 - p_{3S1}) (1 - p_{5S2}) \quad (5.46)$$

$$p_{36} = 1 - (1 - p_{3N1}) (1 - p_{6N2}) \quad (5.47)$$

Na equação (5.39) tem-se que só é possível a ocorrência de uma chamada entre o satélite 1, que pertence ao subsistema 1, e o satélite 4, que pertence ao subsistema 2, se as conexões entre os satélites 1 e S1 (ISL 1-S1) e entre os satélites S2 e 4 (ISL 4-S2) estiverem desbloqueadas. As expressões (5.40) a (5.47) têm interpretações análogas. Assim, nas expressões (5.39) a (5.47), p_{ij} , $i \leq 3 < j \leq 6$, é a probabilidade de bloqueio de chamadas do satélite i , pertencente ao subsistema 1, para o satélite j , pertencente ao subsistema 2; $(1 - p_{iz'})$, $z' \in \{N1, S1\}$, é a probabilidade das chamadas não serem bloqueadas dentro do subsistema 1, ou seja, de elas chegarem até o satélite fictício z' , que é o satélite através do qual a chamada sai/entra no subsistema 1; e $(1 - p_{jz''})$, $z'' \in \{N2, S2\}$, a probabilidade destas chamadas não serem bloqueadas no subsistema 2, ou seja, de elas irem do satélite fictício z'' até o satélite j . Assim, tem-se que:

$$p_{ij} = [1 - (1 - p_{iz'}) (1 - p_{jz''})]$$

Como as chamadas são bidirecionais, através destas expressões, obtêm-se, também, as probabilidades de bloqueio de uma chamada do satélite j , pertencente ao subsistema 2, para o satélite i , pertencente ao subsistema 1.

As equações (5.39) a (5.47) fornecem um valor inicial para as probabilidades $p_{ij}'' = p_{ij}$, $1 \leq i \leq 3 < j \leq 6$, do subsistema 2, que são utilizadas para obter as taxas de chegada de chamadas (equações (5.22) a (5.25)) e as taxas de duração de chamadas (equações (5.34) a (5.37)) deste subsistema. Resolve-se, agora, o subsistema 2, ou seja, obtêm-se as probabilidades limites dos estados do subsistema 2. A seguir, calculam-se as novas probabilidades de bloqueio de chamadas nas conexões do subsistema 2 e nas conexões entre os dois subsistemas. Estes novos valores são, então, usados nas equações (5.8) a

(5.13) e (5.27) a (5.32), fazendo-se $p'_{ij} = p''_{ij}$, para atualizar as taxas de chegada e as taxas de duração das chamadas dos satélites fictícios N1 e S1 do subsistema 1.

O subsistema 1 é, então, resolvido novamente e os valores de p''_{ij} são atualizados e usados nas equações (5.22) a (5.25) e (5.33) a (5.38) para se obter as novas taxas de chegada de chamadas e taxas de duração de chamadas dos satélites fictícios N2 e S2 do subsistema 2. Isto nos conduz a um esquema iterativo no qual os dois subsistemas são resolvidos sucessivamente até que as probabilidades de bloqueio converjam para $p'_{ij} = p''_{ij} = p_{ij}$.

Na Figura 5.2 apresenta-se o algoritmo de decomposição.

1	Iniciar;
2	$h \leftarrow 0$; Fazer $p_{ij} = p^1_{ij} = p^2_{ij} = p_{iz} = p_{iz} = 0$; Ler λ_{ij} ; μ_{ij} $1 \leq i \leq 3 < j \leq 6$;
3	Gerar estados do subsistema 1;
4	Gerar estados do subsistema 2;
5	$h \leftarrow h+1$; <u>Subsistema 1</u> $p^2_{ij} \leftarrow p_{ij}$; Calcular $\lambda_{1N1}, \lambda_{1S1}, \lambda_{2N1}, \lambda_{2S1}, \lambda_{3N1}, \lambda_{3S1}$; $\mu_{1N1}, \mu_{1S1}, \mu_{2N1}, \mu_{2S1}, \mu_{3N1}, \mu_{3S1}$; Calcular as probabilidades limites de cada estado do subsistema 1; Calcular as probabilidades de bloqueio de cada conexão do subsistema 1; Calcular as probabilidades de bloqueio ($p_{14}, p_{15}, p_{16}, p_{24}, p_{25}, p_{26}, p_{34}, p_{35}, p_{36}$); <u>Subsistema 2</u> $p^2_{ij} \leftarrow p_{ij}$; Calcular $\lambda_{N24}, \lambda_{S24}, \lambda_{N25}, \lambda_{S25}, \lambda_{N26}, \lambda_{S26}$; $\mu_{N24}, \mu_{S24}, \mu_{N25}, \mu_{S25}, \mu_{N26}, \mu_{S26}$; Calcular as probabilidades limites de cada estado do subsistema 2; Calcular as probabilidades de bloqueio de cada conexão do subsistema 2; Calcular as probabilidades de bloqueio ($p_{14}, p_{15}, p_{16}, p_{24}, p_{25}, p_{26}, p_{34}, p_{35}, p_{36}$);
6	Repetir o passo 5 até que as probabilidades de bloqueio das conexões entre subsistemas converja;
7	Fim do algoritmo.

Figura 5.2 – Algoritmo de decomposição para uma constelação de seis satélites em uma única órbita

Quando o procedimento iterativo termina, é possível obter as probabilidades de bloqueio entre quaisquer dois satélites. Se ambos os satélites estão no mesmo subsistema, a probabilidade de bloqueio de chamadas entre os satélites é calculada através da equação (3.15) apresentada no Capítulo 3, Seção 3.5. Caso contrário, ou seja, se os satélites estão em subsistemas diferentes, é necessário estimar o valor das probabilidades de bloqueio de chamadas entre estes satélites.

5.4 Probabilidade de bloqueio de chamadas nas conexões entre os dois subsistemas (ZAIM, 2001)

Em Zaim (2001), quando o autor utiliza o seu método de decomposição em uma constelação de satélites com 16 satélites em órbitas diferentes, ele apresenta uma expressão para o cálculo da probabilidade de bloqueio de dois satélites que estão em subsistemas diferentes. Nesta expressão, esta probabilidade é obtida multiplicando-se a probabilidade de bloqueio em cada subsistema por onde estas chamadas passam. Desta forma, são consideradas somente as probabilidades de bloqueio entre o satélite que originou a chamada e a Terra; entre os satélites nos subsistemas, que ficam no percurso por onde passam as chamadas, e os satélites fictícios por onde estas chamadas saem destes subsistemas; e entre o satélite destino da chamada e a Terra. Baseadas na expressão apresentada em Zaim (2001), as probabilidades de bloqueio, para o sistema de seis satélites em estudo neste trabalho, podem ser obtidas como segue:

$$p_{14} = 1 - (1 - p_{11}) (1 - p_{1S1}) (1 - p_{4S2}) (1 - p_{44}) \quad (5.48)$$

$$p_{15} = 1 - (1 - p_{11}) (1 - p_{1N1}) (1 - p_{5N2}) (1 - p_{55}) \quad (5.49)$$

$$p_{16} = 1 - (1 - p_{11}) (1 - p_{1N1}) (1 - p_{6N2}) (1 - p_{66}) \quad (5.50)$$

$$p_{24} = 1 - (1 - p_{22}) (1 - p_{2S1}) (1 - p_{4S2}) (1 - p_{44}) \quad (5.51)$$

$$p_{25} = 1 - (1 - p_{22}) (1 - p_{2N1}) (1 - p_{5N2}) (1 - p_{55}) \quad (5.52)$$

$$p_{26} = 1 - (1 - p_{22}) (1 - p_{2N1}) (1 - p_{6N2}) (1 - p_{66}) \quad (5.53)$$

$$p_{34} = 1 - (1 - p_{33}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{44}) \quad (5.54)$$

$$p_{35} = 1 - (1 - p_{33}) (1 - p_{3S1}) (1 - p_{5S2}) (1 - p_{55}) \quad (5.55)$$

$$p_{36} = 1 - (1 - p_{33}) (1 - p_{3N1}) (1 - p_{6N2}) (1 - p_{66}) \quad (5.56)$$

Na equação (5.48) tem-se que só é possível a ocorrência de uma chamada entre o satélite 1, que pertence ao subsistema 1, e o satélite 4, que pertence ao subsistema 2, se as conexões entre a Terra e o satélite 1 (UDL 1-1), entre os satélites 1 e S1 (ISL 1-S1), entre os satélites S2 e 4 (ISL 4-S2) e entre o satélite 4 e a Terra (UDL 4-4) estiverem desbloqueadas. A probabilidade de estas conexões estarem desbloqueadas é

$$(1 - p_{11}) (1 - p_{1S1}) (1 - p_{4S2}) (1 - p_{44}).$$

Logo,

$$1 - (1 - p_{11}) (1 - p_{1S1}) (1 - p_{4S2}) (1 - p_{44})$$

é a probabilidade de bloqueio de chamadas entre o satélite 1 no subsistema 1 e o satélite 4 no subsistema 2. As equações (5.49) a (5.56) são obtidas de maneira similar.

5.5 Probabilidade de bloqueio de chamadas nas conexões entre os dois subsistemas (ZAIM ET AL., 2003)

Zaim et al. (2003) modelam uma constelação de satélites com 16 satélites em órbitas diferentes e utilizam o método de decomposição proposto em Zaim (2001) para obter as probabilidades de bloqueio deste sistema. Nesta referência, os autores apresentam uma nova expressão para o cálculo da probabilidade de bloqueio de dois satélites que estão em subsistemas diferentes. Nesta expressão, esta probabilidade é obtida multiplicando-se a probabilidade de bloqueio em todas as conexões que estão no percurso destas chamadas, ou seja, são consideradas as probabilidade de bloqueio dentro de cada subsistema e de cada subsistema para o satélites fictício por onde estas chamadas saem do referido subsistema. Baseadas na expressão apresentada em Zaim et al. (2003), as probabilidades de bloqueio, para o sistema de seis satélites em estudo neste trabalho, podem ser obtidas como segue:

$$p_{14} = 1 - (1 - p_{11}) (1 - p_{12}) (1 - p_{23}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{44}) \quad (5.57)$$

$$p_{15} = 1 - (1 - p_{11}) (1 - p_{1N1}) (1 - p_{6N2}) (1 - p_{56}) (1 - p_{55}) \quad (5.58)$$

$$p_{16} = 1 - (1 - p_{11}) (1 - p_{1N1}) (1 - p_{6N2}) (1 - p_{66}) \quad (5.59)$$

$$p_{24} = 1 - (1 - p_{22}) (1 - p_{23}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{44}) \quad (5.60)$$

$$p_{25} = 1 - (1 - p_{22}) (1 - p_{12}) (1 - p_{1N1}) (1 - p_{6N2}) (1 - p_{56}) (1 - p_{55}) \quad (5.61)$$

$$p_{26} = 1 - (1 - p_{22}) (1 - p_{12}) (1 - p_{1N1}) (1 - p_{6N2}) (1 - p_{66}) \quad (5.62)$$

$$P_{34} = 1 - (1 - p_{33}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{44}) \quad (5.63)$$

$$p_{35} = 1 - (1 - p_{33}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{45}) (1 - p_{55}) \quad (5.64)$$

$$p_{36} = 1 - (1 - p_{33}) (1 - p_{23}) (1 - p_{12}) (1 - p_{1N1}) (1 - p_{6N2}) (1 - p_{66}) \quad (5.65)$$

Na equação (5.57) tem-se que só é possível a ocorrência de uma chamada entre o satélite 1, que pertence ao subsistema 1, e o satélite 4, que pertence ao subsistema 2, se as conexões entre a Terra e os satélites 1 (UDL 1-1), entre os satélites 1 e 2 (ISL 1-2), entre os satélites 2 e 3 (ISL 2-3), entre os satélites 3 e S1 (ISL 3-S1), entre os satélites S2 e 4 (ISL 4-S2) e entre os satélites 4 e a Terra (UDL 4-4) estiverem desbloqueadas. A probabilidade de estas conexões estarem desbloqueadas é

$$(1 - p_{11}) (1 - p_{12}) (1 - p_{23}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{44}).$$

Logo,

$$1 - (1 - p_{11}) (1 - p_{12}) (1 - p_{23}) (1 - p_{3S1}) (1 - p_{4S2}) (1 - p_{44})$$

é a probabilidade de bloqueio de chamadas entre o satélite 1 no subsistema 1 e o satélite 4 no subsistema 2. As equações (5.58) a (5.65) são obtidas de maneira similar.

6 RESULTADOS NUMÉRICOS

Neste capítulo analisam-se sistemas de satélites considerando os modelos sem e com decomposição, e o modelo com *handoff*.

Na próxima seção serão apresentados alguns resultados numéricos obtidos utilizando-se o modelo markoviano, apresentado no Capítulo 3, em constelações de satélites em órbita Geoestacionária. Na Seção 6.2, calculam-se as probabilidades de bloqueio de chamadas de sistemas em diferentes altitudes de órbita. Uma comparação entre as probabilidades de bloqueio de chamadas, obtidas ao se utilizar o modelo markoviano apresentado no Capítulo 3 em sistemas sem decomposição, e as obtidas quando estes sistemas são decompostos pelos métodos apresentados no Capítulo 5 é apresentada na Seção 6.3. Na Seção 6.4 analisa-se uma constelação de satélites geoestacionários, em que a distribuição de clientes é heterogênia sobre a superfície terrestre.

Todos os resultados foram obtidos utilizando-se um computador Hp com processador AMD Athlon (tm) 64 3200, 1,79 GHz e 512MB de RAM.

6.1 Sistema Geoestacionário

Nesta seção, analisam-se constelações de satélites em órbita geoestacionária, usando a modelagem markoviana apresentada no Capítulo 3. Consideram-se, inicialmente, constelações com dois e quatro satélites, com $C_{UDL} = C_{ISL}$ assumindo os valores 3, 5 e 10; e constelações com seis satélites, $C_{UDL} = C_{ISL}$ assumindo os valores 2, 3 e 5. Na Tabela 6.1, apresentam-se a quantidade de estados e transições para as constelações consideradas. Para obtenção destes resultados, em todos os sistemas apresentados na Tabela 6.1, admitiram-se $\lambda_{ij} = 1$ e $\mu_{ij} = 2$, $1 \leq i \leq j \leq 6$; salienta-se que a quantidade de estados e de transições só depende do número de satélites e de canais UDL e ISL, ou seja, independe dos valores de λ e μ .

Tabela 6.1 – Número de estados e transições

Número de satélites	Número de canais para cada satélite ($C_{UDL} = C_{ISL}$)	Número de estados	Número de transições
2	3	10	28
	5	28	102
	10	146	660
4	3	722	4.928
	5	14.138	134.970
	10	1.960.575	25.712.940
6	2	5.688	44.732
	3	128.152	1.454.898
	5	34.603.358	*

Nota: * significa que não foi possível gerar resultados por falta de capacidade computacional.

Observa-se, na Tabela 6.1, que ocorre um aumento exponencial na quantidade de estados e transições quando o a quantidade de satélites e/ou a capacidade de conexão das constelações de satélites aumentam, sendo computacionalmente oneroso gerar as transições para um sistema com seis satélites, cinco canais ISL e cinco canais UDL. Dessa forma, para analisar o desempenho dos sistemas de satélites utilizando a modelagem proposta neste trabalho, serão considerados, neste capítulo, sistemas de quatro satélites com C_{UDL} e C_{ISL} assumindo os valores 3, 5 e 10 e sistemas de seis satélites com C_{UDL} e C_{ISL} assumindo os valores 2 e 3.

Na Figura 6.1 mostra-se como a probabilidade de bloqueio do sistema varia em função da taxa de chegada de chamadas para uma constelação de seis satélites, em que $C_{UDL} = C_{ISL} = 3$, $\mu = 2$ e λ varia entre 0,01 e 0,5.

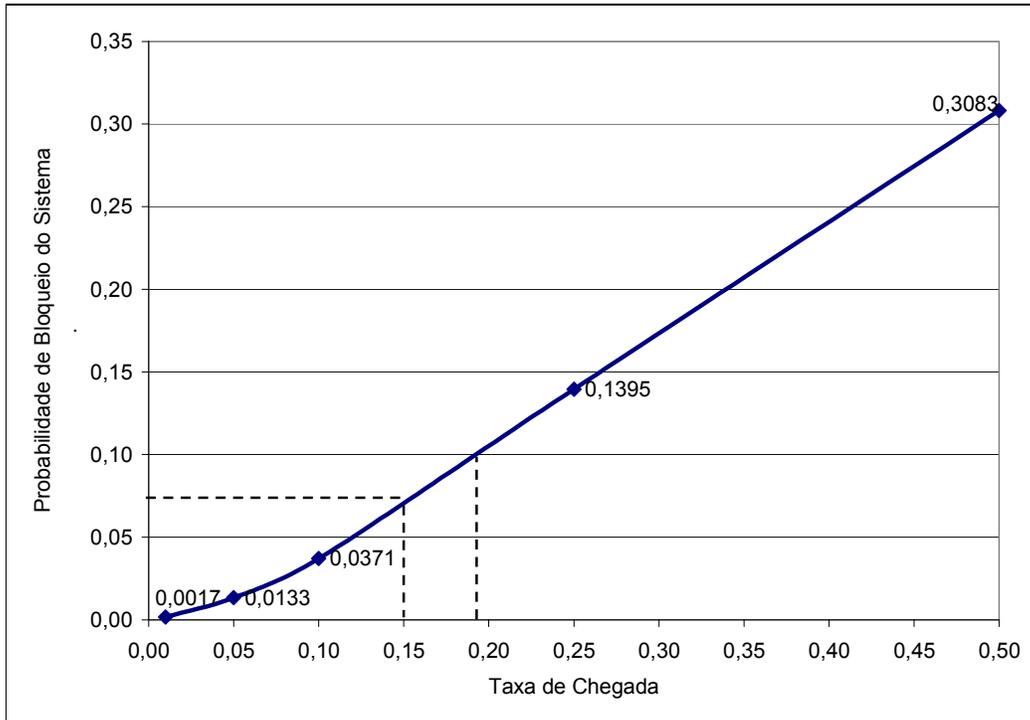


Figura 6.1 – Variação da probabilidade de bloqueio do sistema em função da taxa de chegada de chamadas (6 satélites; $C_{UDL} = C_{ISL} = 3$; $\mu = 2$).

Para ilustrar a obtenção das medidas de desempenho, considera-se uma constelação de seis satélites com $\mu_{ij} = 2$, $1 \leq i \leq j \leq 6$, e $C_{UDL} = C_{ISL}$ assumindo valores 2 e 3. No sistema com $C_{UDL} = C_{ISL} = 3$, considerou-se $\lambda_{ij} = 0,15$. Com esta taxa de chegada, a probabilidade de bloqueio do sistema é menor do que 10%, mais especificamente 7%, como mostrado na Tabela 6.1, o que foi considerada, neste trabalho, como uma probabilidade de bloqueio aceitável. No sistema com $C_{UDL} = C_{ISL} = 2$ e $\lambda_{ij} = 0,15$, a probabilidade de bloqueio do sistema fica em torno de 22%, mas optou-se por manter a mesma taxa de chegada para os dois sistemas.

Para ambos os sistemas, foram calculados o número médio de chamadas entre dois satélites e o número médio de canais usados nas conexões físicas do sistema, que são apresentados, respectivamente, nas Tabelas 6.2 e 6.3.

Tabela 6.2 – Número médio de chamadas nas conexões lógicas

Conexões	Número médio de chamadas	
	$C_{UDL} = C_{ISL} = 2$	$C_{UDL} = C_{ISL} = 3$
1 – 1	0,0512	0,0664
1 – 2	0,0612	0,0713
1 – 3	0,0599	0,0711
1 – 4	0,0589	0,0709
1 – 5	0,0606	0,0713
1 – 6	0,0616	0,0714
2 – 2	0,0512	0,0664
2 – 3	0,0616	0,0714
2 – 4	0,0606	0,0713
2 – 5	0,0589	0,0709
2 – 6	0,0599	0,0711
3 – 3	0,0511	0,0664
3 – 4	0,0622	0,0716
3 – 5	0,0612	0,0715
3 – 6	0,0587	0,0708
4 – 4	0,0510	0,0664
4 – 5	0,0624	0,0716
4 – 6	0,0612	0,0715
5 – 5	0,0510	0,0664
5 – 6	0,0622	0,0716
6 – 6	0,0511	0,0664

Tabela 6.3 – Número médio de canais usados nas conexões físicas

Conexões	Número médio de canais	
	$C_{UDL} = C_{ISL} = 2$	$C_{UDL} = C_{ISL} = 3$
1 – 1	0,4046	0,4889
2 – 2	0,4046	0,4889
3 – 3	0,4059	0,4892
4 – 4	0,4074	0,4897
5 – 5	0,4074	0,4897
6 – 6	0,4059	0,4892
1 – 2	0,3576	0,4261
2 – 3	0,2998	0,3556
3 – 4	0,2430	0,2853
4 – 5	0,1848	0,2145
5 – 6	0,2430	0,2853
1 – 6	0,2998	0,3556

Como todas as conexões com a Terra são feitas através das conexões UDL, as quantidades de canais usados nestas conexões são sempre maiores que nas conexões ISL's, como pode ser observado nas Tabela 6.2 e 6.3. Pode-se observar também, nestas tabelas, que o número de canais usados na conexão ISL entre os satélites 4 e 5 é menor do que nas outras conexões. Isto porque nesta conexão só ocorrem ligações entre estes dois satélites, enquanto as outras conexões são compartilhadas por mais do que dois satélites.

Para analisar o comportamento das probabilidades de bloqueio nas conexões do sistema em função da capacidade dessas conexões, elaboraram-se os gráficos das Figura 6.2 e 6.3. Nestes gráficos são apresentados, respectivamente, a variação das probabilidades de bloqueio em cada conexão de uma constelação de quatro satélites, em uma única órbita, em função da variação da capacidade C_{UDL} fixando o valor de C_{ISL} , e em função da variação da capacidade C_{ISL} fixando o valor de C_{UDL} .

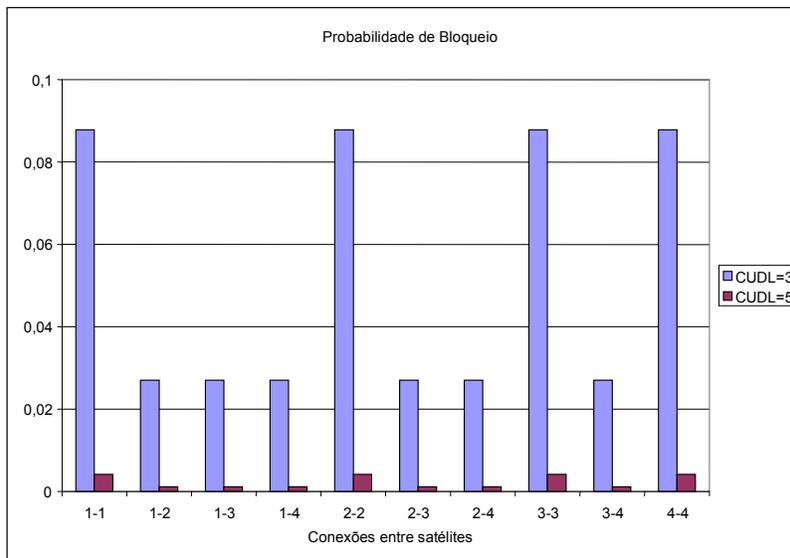


Figura 6.2 – Variação da probabilidade de bloqueio em cada conexão (4 satélites; $C_{ISL} = 10$; $\lambda = 0,15$; $\mu = 2$)

Na Figura 6.2 observa-se que, quando $C_{ISL} = 10$ e $C_{UDL} = 3$, tem-se uma probabilidade de aproximadamente 9% de se bloquear as chamadas nas conexões UDL (1-1, 2-2, 3-3,

4-4) e uma probabilidade de aproximadamente 3% de se bloquear as chamadas nas conexões ISL's (1-2, 1-3, 1-4, 2-3, 2-4 e 3-4). Quando se mantém $C_{ISL} = 10$ e aumenta-se o valor de C_{UDL} , ou seja, $C_{UDL} = 5$, a probabilidade de bloqueio de chamadas nas conexões UDL's diminui para aproximadamente 0,4% e nas conexões ISL para aproximadamente 0,1%.

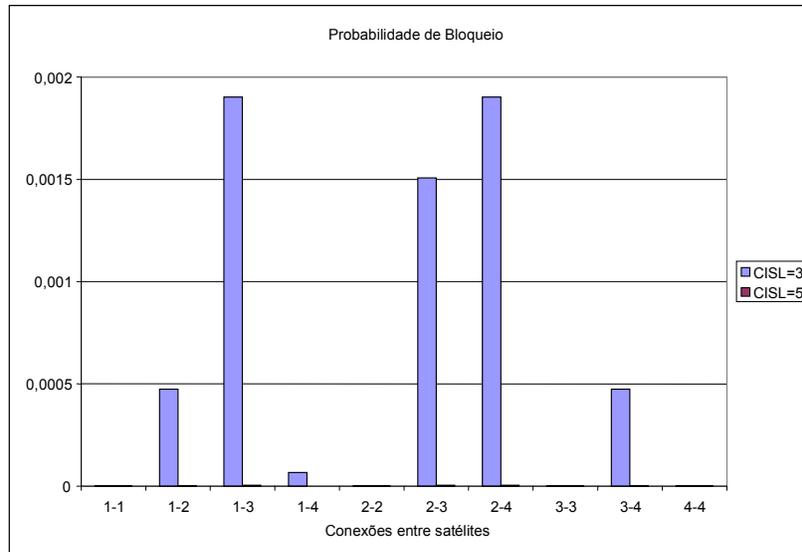


Figura 6.3 – Variação da probabilidade de bloqueio em cada conexão (4 satélites; $C_{UDL} = 10$; $\lambda = 0,15$; $\mu = 2$)

Na Figura 6.3 observa-se que, quando se faz o contrário, $C_{ISL} = 3$ e $C_{UDL} = 10$, a probabilidade de bloqueio nas conexões ISL's fica abaixo de 0,2% e muito próximo de zero nas conexões UDL's. E, para $C_{UDL} = 10$, se C_{ISL} é aumentado, assumindo valor 5, a probabilidade de bloqueio é praticamente nula em todas as conexões.

Este comportamento em torno das variações de C_{UDL} e C_{ISL} mostra que as conexões *up-and-downlink* têm um efeito de gargalo maior do que as conexões ISL's, bloqueando, assim, o funcionamento do sistema. Isto se deve ao fato de que todas as chamadas passam, necessariamente, por estas conexões, antes de passarem pelas conexões ISL's.

6.2 Sistema com *handoff* ou não-geoestacionário

Considere uma constelação de seis satélites com $C_{ISL} = C_{UDL} = 3$, $\mu_{ij} = 2$, $\lambda_{ij} = 0,15$, $1 \leq i \leq j \leq 6$. Na Tabela 6.4 pode-se observar a variação da probabilidade de bloqueio do sistema em função da taxa de *handoff* α , ou seja, a proporção de transferência de chamadas para o próximo satélite devido ao *handoff*.

Como α é diretamente proporcional à velocidade do satélite, quanto maior α , mais *handoffs* ocorrerão e maior será a probabilidade de bloqueio de chamadas no sistema, como pode ser visto na Tabela 6.4. Em uma órbita geoestacionária, $\alpha = 0$, ou seja, a velocidade do satélite é a mesma que a velocidade de rotação da Terra, e não são realizadas operações de *handoff*.

Tabela 6.4 – Variação da probabilidade de bloqueio do sistema em função da taxa de *handoff*

α	Probabilidade de bloqueio do sistema
0,00	0,0680
0,25	0,0950
0,50	0,1120
0,75	0,1227
1,00	0,1296
1,40	0,1360

Sejam a área da superfície terrestre $510.000.000 \text{ km}^2$ e a velocidade angular de satélites do tipo LEO aproximadamente 25.000 km/h , a área de cobertura dos satélites, A , é aproximadamente $25.500.000 \text{ km}^2$ (5% da superfície terrestre). Considerando a área de cobertura dos satélites com uma forma elíptica de raios 910 km e 8920 km , tem-se que em um tempo $\Delta t = 0,08 \text{ horas}$ (5 minutos) o satélite varrerá uma área de aproximadamente $\Delta A = 2.860.000 \text{ km}^2$.

Utilizando os dados do parágrafo anterior e considerando uma aproximação da taxa de *handoff* $\alpha = \Delta A / (A \cdot \Delta t)$, tem-se que $\alpha \cong 1,40$. Pela Tabela 6.4, para este valor de α , a probabilidade de bloqueio do sistema é igual a 0,1360.

6.3 Sistema decomposto

Nesta seção, consideram-se os mesmos sistemas apresentados na Seção 6.1, ou seja, dois sistemas de seis satélites, com $\lambda_{ij} = 0,15$, $\mu_{ij} = 2$, $1 \leq i \leq j \leq 6$, $C_{ISL} = C_{UDL}$ assumindo valores 2 e 3. O objetivo é analisar o desempenho dos métodos de decomposição, apresentados no Capítulo 5, quando comparados com os resultados do modelo sem decomposição.

Nas Tabelas 6.5 e 6.6, apresentam-se as probabilidades de bloqueio em cada conexão, obtidos nos modelos sem e com decomposição.

Tabela 6.5 – Probabilidade de bloqueio em um sistema de seis satélites sem e com decomposição, com $C_{ISL} = C_{UDL} = 2$

Conexão	Probabilidade de Bloqueio			
	Modelo sem decomposição	Modelo com decomposição		
		Zaim (2001)	Zaim et al. (2003)	Proposta deste trabalho
1-1	0,3179	0,5246	0,5323	0,5083
1-2	0,1847	0,3471	0,3591	0,2966
1-3	0,2012	0,3531	0,3632	0,3184
1-4	0,2140	0,8534	0,9407	0,3439
1-5	0,1914	0,8513	0,9060	0,3237
1-6	0,1783	0,8484	0,8558	0,3066
2-2	0,3179	0,5238	0,5251	0,5055
2-3	0,1783	0,3455	0,3585	0,2901
2-4	0,1914	0,8517	0,9060	0,3183
2-5	0,2140	0,8550	0,9388	0,3515
2-6	0,2012	0,8521	0,9061	0,3352
3-3	0,3186	0,5247	0,5323	0,5106
3-4	0,1704	0,8482	0,8557	0,2917
3-5	0,1839	0,8508	0,9058	0,3088
3-6	0,2176	0,8536	0,9407	0,3547

Tabela 6.5 – Continuação

	Probabilidade de Bloqueio			
	Modelo sem decomposição	Modelo com decomposição		
		Zaim (2001)	Zaim et al. (2003)	Proposta deste trabalho
4-4	0,3196	0,5248	0,5324	0,5131
4-5	0,1685	0,3426	0,3573	0,2792
4-6	0,1839	0,3495	0,3619	0,3005
5-5	0,3196	0,5242	0,5252	0,5089
5-6	0,1704	0,3438	0,3579	0,2813
6-6	0,3186	0,5247	0,5324	0,5085

Tabela 6.6 – Probabilidade de bloqueio em um sistema de seis satélites sem e com decomposição, com $C_{ISL} = C_{UDL} = 3$

Conexão	Probabilidade de Bloqueio			
	Modelo sem decomposição	Modelo com decomposição		
		Zaim (2001)	Zaim et al. (2003)	Proposta deste trabalho
1-1	0,1141	0,1718	0,1727	0,1717
1-2	0,0495	0,0650	0,0636	0,0729
1-3	0,0525	0,0677	0,0663	0,0773
1-4	0,0545	0,3633	0,4367	0,0838
1-5	0,0492	0,3604	0,3982	0,0788
1-6	0,0474	0,3590	0,3596	0,0761
2-2	0,1141	0,1718	0,1718	0,1715
2-3	0,0474	0,0634	0,0622	0,0701
2-4	0,0492	0,3601	0,3978	0,0764
2-5	0,0545	0,3638	0,4358	0,0861
2-6	0,0525	0,3624	0,3996	0,0834
3-3	0,1142	0,1720	0,1729	0,1720
3-4	0,0457	0,3578	0,3586	0,0713
3-5	0,0471	0,3587	0,3968	0,0733
3-6	0,0556	0,3645	0,4377	0,0880
4-4	0,1144	0,1722	0,1731	0,1726
4-5	0,0452	0,0616	0,0607	0,0672
4-6	0,0471	0,0636	0,0629	0,0701
5-5	0,1144	0,1721	0,1721	0,1722
5-6	0,0457	0,0621	0,0612	0,0678
6-6	0,1142	0,1720	0,1729	0,1720

Os resultados das Tabelas 6.5 e 6.6 mostram que, com o aumento da quantidade de canais nos sistemas, as diferenças entre as probabilidades de bloqueio, calculadas pelos métodos de decomposição apresentados neste trabalho, diminuem em relação às obtidas para os sistemas não decompostos. Isto pode indicar que à medida que se aumenta a quantidade de canais UDL e ISL do sistema, os métodos de decomposição tendem a aproximar melhor os modelos sem decomposição, mas são necessários mais testes para se comprovar este resultado.

Pode-se observar, também, que nas conexões 1-4, 1-5, 1-6, 2-4, 2-5, 2-6, 3-4, 3-5 e 3-6, referentes às ligações entre os subsistemas 1 e 2, os resultados obtidos para as probabilidades de bloqueio, utilizando-se as equações propostas pelo presente trabalho, estão mais próximos dos resultados encontrados para os modelos sem decomposição, do que as probabilidades de bloqueio dos sistemas decompostos obtidas ao se utilizar as equações apresentadas por Zaim (2001) e Zaim et al. (2003).

Para um melhor entendimento dos resultados mostrados nas Tabelas 6.5 e 6.6 foram construídos os gráficos apresentados nas Figuras 6.4 e 6.5, respectivamente.

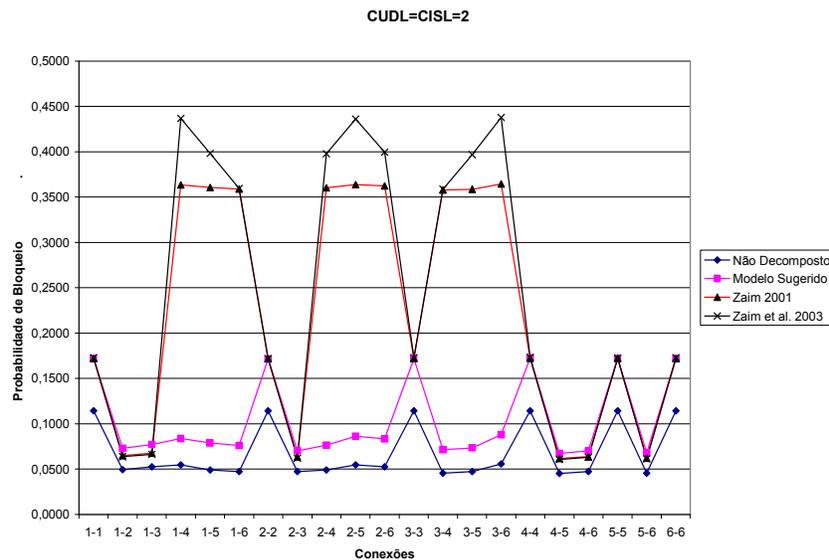


Figura 6.4 – Probabilidade de bloqueio em cada conexão do sistema para cada método de decomposição, considerando $C_{ISL} = C_{UDL} = 2$.

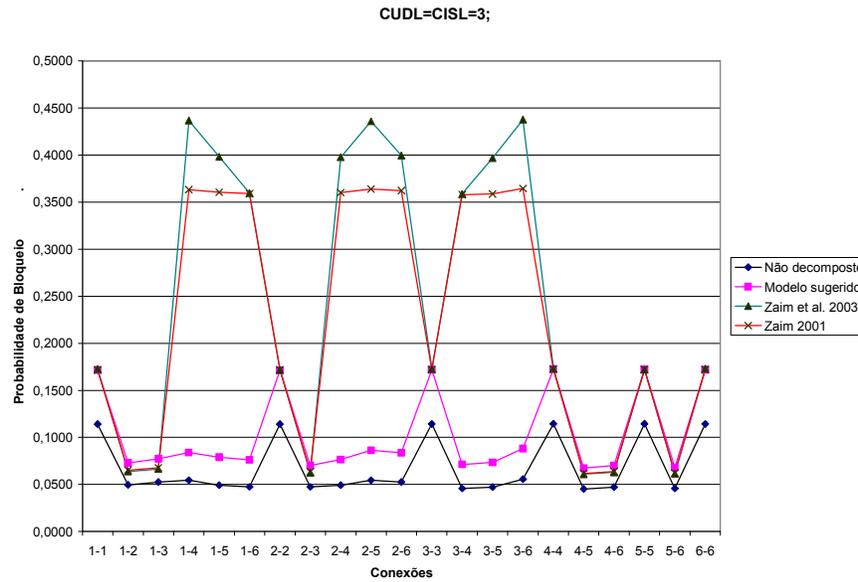


Figura 6.5 – Probabilidade de bloqueio em cada conexão do sistema para cada método de decomposição, considerando $C_{ISL} = C_{UDL} = 3$.

6.4 Sistema com distribuição heterogênea de clientes sobre a superfície terrestre

Nesta seção, considera-se um sistema composto por seis satélites igualmente espaçados em uma única órbita geoestacionária, no qual se admite uma distribuição heterogênea de clientes sobre a superfície terrestre, variando-se a taxa de chegada de chamadas em cada conexão.

Para gerar as taxas de chegada de chamada em cada conexão, considerou-se $\lambda_{ii} = 0,15$, para $i = \{1, 2, 3\}$, $\lambda_{ii} = 0,10$ para $i = \{4, 6\}$, e $\lambda_{ii} = 0,05$ para $i = 5$. As outras taxas de chegada foram obtidas usando-se a equação (6.1):

$$\lambda_{ij} = \frac{\lambda_{ii} + \lambda_{jj}}{2^{(n_{ISL} + 1)}} \quad (6.1)$$

em que n_{ISL} é a quantidade de conexões ISL utilizadas para completar a chamada.

Na equação (6.1), considerou-se que as taxas de chegada diminuem à medida que a distância entre os satélites, número de conexões ISL, aumenta. As taxas de chegada obtidas para todas as conexões UDL e ISL são apresentadas na Tabela 6.7.

Tabela 6.7 – Taxas de chegada de chamadas em cada conexão para uma distribuição heterogênea de clientes sobre a superfície terrestre

Conexão <i>i-j</i>	λ_{ij}
1 – 1	0,150000
1 – 2	0,075000
1 – 3	0,037500
1 – 4	0,015625
1 – 5	0,025000
1 – 6	0,062500
2 – 2	0,150000
2 – 3	0,075000
2 – 4	0,031250
2 – 5	0,012500
2 – 6	0,031250
3 – 3	0,150000
3 – 4	0,062500
3 – 5	0,025000
3 – 6	0,015625
4 – 4	0,100000
4 – 5	0,037500
4 – 6	0,025000
5 – 5	0,050000
5 – 6	0,037500
6 – 6	0,100000

No sistema apresentado na Seção 6.1, para uma distribuição homogênea de chamadas sobre a superfície terrestre, ou seja, com taxas de chegada de chamadas iguais em todas as conexões, considerou-se que as chamadas chegavam ao seu destino utilizando uma rota de menor caminho. Como no sistema considerado nesta seção, as conexões UDL 1, 2 e 3 têm taxas de chegada de chamadas maiores que as conexões UDL 4, 5 e 6, optou-se por rotear as chamadas pelas conexões menos congestionadas (rota de menor fluxo). Na rota de menor fluxo adotada, considera-se que as chamadas entre os satélites 1 e 4 passam pelos satélites 1, 6, 5 e 4; que as chamadas entre os satélites 3 e 6 passam pelos

satélites 3, 4, 5 e 6; e que as chamadas entre os satélites 2 e 5 passam pelos satélites 2, 3, 4 e 5. Todas as outras chamadas seguem o rota de menor caminho, apresentado no Capítulo 3.

Na Tabela 6.8 apresenta-se a variação da probabilidade de bloqueio em função da capacidade C_{ISL} e da taxa de duração de chamadas em cada conexão para as chamadas que iniciam e terminam no mesmo satélite, ou seja chamadas que utilizam somente os canais UDL, e chamadas que utilizam uma, duas ou três conexões ISL para serem realizadas. Para todos os resultados mostrados nesta tabela, considerou-se $C_{UDL} = 3$.

Tabela 6.8 – Probabilidade de bloqueio de chamadas considerando rotas diferentes

PROBABILIDADE DE BLOQUEIO				
Rota de menor caminho				
Número de conexões ISL utilizadas	3ISL $\mu = 2$	3ISL $\mu = 0,5$	2ISL $\mu = 0,5$	1ISL $\mu = 0,5$
0	0,0643	0,2365	0,2339	0,2161
1	0,0109	0,1095	0,1233	0,2443
2	0,0104	0,1069	0,1346	0,3502
3	0,0103	0,1078	0,1587	0,4700
Rota de menor fluxo				
Número de conexões ISL utilizadas	3ISL $\mu = 2$	3ISL $\mu = 0,5$	2ISL $\mu = 0,5$	1ISL $\mu = 0,5$
0	0,0643	0,2366	0,2343	0,2166
1	0,0109	0,1094	0,1219	0,2415
2	0,0104	0,1066	0,1330	0,3509
3	0,0102	0,1063	0,1443	0,4196

Pode-se observar na Tabela 6.8 que, no sistema considerado, ao se utilizar, para as chamadas que passam por três conexões, a rota de menor fluxo, a probabilidade de bloqueio destas chamadas diminuem consideravelmente, em relação ao sistema que usa a rota de caminho mínimo. Isto é mais evidente em sistemas sobrecarregados, como no sistema com as taxas de chegada mostradas na Tabela 6.7, $C_{UDL} = 3$, $C_{ISL} = 1$ e $\mu = 0,5$.

7 CONCLUSÃO

Neste trabalho, apresentou-se um modelo markoviano a tempo contínuo para uma constelação de seis satélites de comunicação em uma única órbita. Inicialmente, considerou-se uma constelação geoestacionária, e, posteriormente, os satélites foram considerados móveis em relação à superfície terrestre, ou seja, o caso da órbita LEO. O objetivo foi descrever e analisar o funcionamento desta constelação. Tomou-se como ponto de partida o modelo proposto em Zaim (2001).

Entre as contribuições do presente trabalho incluem-se as equações para obtenção das seguintes medidas de desempenho para analisar o funcionamento do sistema em estudo: número médio de chamadas entre dois satélites, número médio de canais usados, taxa de rejeição de chamadas no sistema e em cada conexão e taxa efetiva de chamadas em cada conexão. Para a obtenção destas medidas, utilizaram-se as probabilidades estacionárias do sistema, também denominadas probabilidades limite, calculadas na cadeia de Markov a tempo contínuo através da solução na forma de produto, apresentada no Capítulo 2.

A mais importante restrição ao modelo proposto acontece quando se aumentam o número de satélites e/ou suas respectivas capacidades de conexões, pois o espaço de estados do sistema cresce exponencialmente, ficando computacionalmente oneroso gerar seus estados, transições e as medidas de desempenho. Para sistemas com mais de seis satélites, são necessários métodos que tratem cadeias de Markov com espaço de estados muito grande. Neste trabalho, utilizou-se, para este fim, o método de decomposição proposto por Zaim (2001).

O método de decomposição empregado consiste em dividir o sistema de seis satélites em dois subsistemas de três satélites cada um e utilizar o modelo sugerido para o sistema sem decomposição em cada um dos subsistemas, isoladamente. Os resultados de cada subsistema são combinados através de um algoritmo de agregação iterativo e, a cada iteração, calculam-se as probabilidades de bloqueio de chamadas nas conexões

entre os dois subsistemas até que elas convirjam para um mesmo valor. As probabilidades de bloqueio das chamadas são calculadas quando o sistema está em regime estacionário, ou seja, utilizando-se as probabilidades limites do sistema. Para o cálculo destas probabilidades são necessárias as taxas de chegada e as taxas de duração das chamadas.

Outras contribuições obtidas com o presente trabalho incluem o desenvolvimento de equações para cálculo das taxas duração de chamadas e para o cálculo das probabilidades de bloqueio de chamadas realizadas entre satélites que estão em subsistemas diferentes. Os resultados numéricos obtidos, para os dados de entrada considerados, mostraram que as probabilidades de bloqueio, calculadas através das equações propostas por este trabalho, ficaram mais próximas das obtidas para o sistema sem decomposição, do que as equações propostas por Zaim (2001) e Zaim et al. (2003).

7.1 Propostas de trabalhos futuros

Propõem-se as seguintes sugestões para trabalhos futuros:

1. A modelagem de processos de demandas reais, onde as demandas não são uniformemente distribuídas no tempo e no espaço. No desenvolvimento deste trabalho foi considerada uma distribuição homogênea de clientes sobre a superfície terrestre, o que não condiz com a realidade, já que a população mundial que usa sistemas de satélites de comunicação é mais concentrada nas áreas de continente. Zaim et al. (2003) sugere considerar, para este fim, cadeias de Markov periódicas.
2. Considerar outras distribuições para os tempos de duração das chamadas. Sumter (2003), por exemplo, já propõe modelar estes tempos por distribuições do tipo-fase.

3. Um refinamento sobre a modelagem de sistemas com *handoff*, que leve em conta a movimentação dos usuários dentro da área de cobertura de cada satélite ou constelações de satélites que se movem entre si, ou seja, não necessariamente sobre a mesma órbita. Nesta modelagem pode-se, também, dar maior prioridade aos *handoffs* em detrimento das chamadas novas, com o objetivo de não interromper as chamadas que já estão em andamento.
4. O estudo de outros métodos de decomposição que permitam a formulação e tratamentos de redes com um grande número de satélite e que encontrem resultados mais precisos do que os obtidos pelos métodos apresentados.
5. O uso de modelos similares ao proposto para determinar o melhor roteamento na constelação de chamadas entre usuários distantes entre si. Note-se que no modelo apresentado é suposto um roteamento pré-fixado.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLEN, A. O. **Probability, statistics, and queueing theory: with computer science applications**. 1. ed. London: Academic Press, Inc., 1978. 390 p.

ALTMAN, E. **Applications of Markov decision processes in communication networks: a survey**. Valbonne, France: Institut National de Recherche en Informatique et en Automatique, INRIA, 2000. 51p. (Rapport de Recherche N° 3984)

ASMUSSEN, S. **Applied probability and queues**. 1. ed. Chichester, England: John Wiley & Sons Ltd., 1987. 318 p.

BEM, D. J.; WIECKOWSKI, T. W.; ZIELINSKI, R. J. Broadband Satellite Systems, **IEEE Communications Surveys & Tutorials**, v. 3, n. 1, p.2-15, First Quarter 2000. disponível em: <<http://www.comsoc.org/livepubs/surveys/public/1q00issue/pdf/Zielinski.pdf>>. Acesso em: 23 setembro 2007.

BOLCH, G.; GREINER, S.; MEER, H.; TRIVEDI, K. S. **Queueing networks and Markov chains: modeling and performance evaluation with computer**. 1. ed. New York: John Wiley & Sons Ltd., 1998. 726 p.

ÇINLAR, E. **Introduction to stochastic processes**. 1. ed. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1975. 402 p.

CLARKE, A. C. Extra Terrestrial Relays. **Wireless World**, p. 305-308, 1945. disponível em: <<http://lakdiva.org/clarke/1945ww/>>. Acesso em: 23 setembro 2007.

COOPER, R. B. **Introduction to queueing theory**. 2. ed. London, United Kingdom: Edward Arnold, 1981. 347 p.

ELBERT, B. R. **The satellite communication applications handbook**. 2. ed. Boston-London: Artech House, Inc., 2004. 532 p.

FELIX, L. E. M., **Predição da indisponibilidade por chuvas em redes VSAT no Brasil**, 2004. 96 p. Dissertação de mestrado – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, Rio de Janeiro, Rio de Janeiro, Brasil, 2004.

GELENBE, E.; PUJOLLE, G. **Introduction to queueing networks**. 1. ed. Paris: John Wiley & Sons Ltd., 1987. 177 p.

HA, T. T. **Digital satellite communications**. 2. ed. Singapore: McGraw-Hill, 1990. 641 p.

HENDERSON, T. R.; KATZ, R. H., Network simulation for LEO satellite networks. In: AIAA INTERNATIONAL COMMUNICATION SATELLITE SYSTEMS CONFERENCE AND EXHIBIT, 18., 2000, Oakland, CA, EUA. **Collection of Technical Papers**. Oakland: AIAA, 2000. v. 2.

KAFFEETY, H. D.; MEYER, C. D.; STEWART, W. J. **A general framework for iterative aggregation/disaggregation methods**. Raleigh, North Carolina: North Carolina State University, 1996. 22 p.

KELLY, F. P. **Reversibility and stochastic networks**. 1. ed. London, United Kingdom: John Wiley & Sons Ltd., 1979. 230 p.

MARAL, G.; BOUSQUET, M. **Satellite communications systems: systems, techniques and technology**. 2. ed. London, England: John Wiley & Sons Ltd, 1993. 688p.

MAREK, I.; PULTAROVA, I. **A note on local and global convergence analysis of iterative aggregation-disaggregation methods**. Prague, Czech Republic: Czech Technical University in Prague, 2005. 14 p.

MELSA, J. L. **An introduction to probability and stochastic processes**. 1. ed. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1973. 403 p.

PINHEIRO, G. **Teoria de filas e sistemas de comunicação**. Rio de Janeiro: UERJ, 2004. Notas de aula. Disponível em <<http://www.lee.eng.uerj.br/~gil/Filas.pdf>> . Acesso em: 19 fevereiro 2008. p.59

PUTERMAN, M.L. **Markov decision processes: discrete stochastic dynamic programming**, 1. ed. New York: John Wiley and Sons, Inc., 1994. 672 p.

ROSS, S. M. **Stochastic processes**, 1. ed. New York: John Wiley & Sons, Inc., 1982. 309 p.

SHERIFF, R. E.; HU, Y. F. **Mobile satellite communications networks**. 1. ed. London, England: John Wiley & Sons Ltd., 2001. 368 p.

STEWART, W. J. **Introduction to the numerical solution of Markov chains**. 1. ed. , Princeton , New Jersey: Princeton University Press, 1994. 539 p.

STEWART, W. J.; WU, W. **Numerical experiments with iteration and aggregation for markov chains**. Raleigh, North Carolina: North Carolina State University, 1996. 33 p.

STRELEN, J. C. **Approximate disaggregation-aggregation solutions for general queueing networks**. Bonn, Germany: Rheinische Friedrich–Wilhelms-Universität Bonn, 1997. 6 p.

STRELEN, J. C.; BARK, B.; BECKER, J.; JONAS, V. **Analysis of queueing networks with blocking using a new aggregation technique**. Bonn, Germany: Rheinische Friedrich–Wilhelms-Universität Bonn, 1997. 20 p.

SUMTER, B. R. **Optimal replacement policies for satellite constellations**, 2003. 88 p. Tese (Mestrado em Pesquisa Operacional) – Air Force Institute of Technology, Ohio, United States, 2003.

TIJMS, H. C. **Stochastic models: an algorithmic approach**. 1. ed. New York: John Wiley & Sons, 1994. 390 p.

USAHA, W.; BARRIA, J. Markov decision theory framework for resource allocation in LEO satellite constellations. **IEEE Proc. Commun**, v. 149, n. 5, p. 270-276, October 2002.

WOLFF, R. W. **Stochastic modeling and the theory of queues**. 1. ed. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1989. 556 p.

WOOD, L., **Internetworking with satellite constellations**, 2001. 208 p. Tese (Doutorado em Filosofia) - Centre for Communication Systems Research, School of Electronics, Computing and Mathematics, University of Surrey, Guildford, United Kingdom, 2001.

ZAIM, A. H., **Computing call-blocking probabilities in LEO satellite networks**, 2001. 91p. Tese (Doutorado em Filosofia) – Electrical and Computer Engineering School, North Carolina State University, Raleigh, United States, 2001.

ZAIM, A. H.; PERROS, H. G.; ROUSKAS, G. N. Computing call-blocking probabilities in LEO satellite constellations. **IEEE Trans. Veh. Technol.**, v. 52, n. 3, p. 622-636, May 2003.

A APÊNDICE A – IMPLEMENTAÇÃO COMPUTACIONAL

Apresentam-se a seguir os códigos dos programas desenvolvidos em linguagem C++ para a construção dos modelos apresentados nos Capítulos 3, 4 e 5 e para a obtenção dos resultados numéricos apresentados no Capítulo 6. Este apêndice está dividido em três seções:

A.1 Código da implementação do modelo markoviano do sistema de seis satélites geostacionários apresentado no Capítulo 3.

A.2 Código da implementação do modelo markoviano do sistema de seis satélites não-geostacionários (com *handoff*) apresentado no Capítulo 4.

A.3 Código da implementação do modelo markoviano do sistema de seis satélites geostacionários com decomposição, utilizando as equações de probabilidade de bloqueio entre subsistemas sugeridas por Zaim (2001); Zaim et al. (2003); e sugeridas por este trabalho, apresentados no Capítulo 5.

A.1 Código da implementação do modelo markoviano do sistema de seis satélites geostacionários apresentado no Capítulo 3

```
#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

struct dtSis {
    int v[21];
};

//declaracao das funcoes
void gera(void);
void leConf(void);
void criaTabFat(int n);
bool valido(int, int, int,
            int, int, int, int, int, int, int, int, int);
void proBlo(void);
bool svalido(dtSis*);
void leTaxas(void);
void desempenho(void);

//declaracao das variaveis globais
int* tabFat = 0;
int cudl, cisl;
//vetores com as taxas de chegada e saida do sistema completo
double chegada[21], saida[21];
double proBloSis[21];

//medida de desempenho
double medcha[21], medcan[12], rejcha[21], taxaAtend[21];
double rejTotal, proBSis, cheTotal;

int main()
{
    int op;
    leConf();
    leTaxas();
    do {
        cout << "\nSelecione\n1 - Gerar estados do sistema";
        cout << "\n2 - Calcular probabilidades de bloqueio";
        cout << "\n3 - Calcular medidas de desempenho\n4 - Sair\n";
        cin >> op;
        switch (op) {
            case 1:
                gera();
                break;
            case 2:
                proBlo();
                break;
            case 3:
                desempenho();
                break;
        }
    } while (op != 4);
    if (tabFat != 0) {
        delete tabFat;
    }
}
```

```

        return 0;
    }
    //gera os estados do sistema
    void gera(void) {
        fstream* f = NULL;
        if ((f = new fstream("completo.txt", ios::out | ios::trunc)) == 0)
        {
            cout << "\nErro ao abrir arquivo";
            return;
        }
        int n11, n12, n13, n14, n15, n16, n22, n23, n24, n25, n26, n33, n34,
            n35, n36, n44, n45, n46, n55, n56, n66;
        int mudl;
        mudl = cudl / 2;
        for (n11 = 0; n11 <= mudl; n11++) {
            for (n12 = 0; n12 <= cisl; n12++) {
                for (n13 = 0; n13 <= cisl; n13++) {
                    for (n14 = 0; n14 <= cisl; n14++) {
                        for (n15 = 0; n15 <= cisl; n15++) {
                            for (n16 = 0; n16 <= cisl; n16++) {
                                for (n22 = 0; n22 <= mudl; n22++) {
                                    for (n23 = 0; n23 <= cisl; n23++) {
                                        for (n24 = 0; n24 <= cisl; n24++) {
                                            for (n25 = 0; n25 <= cisl; n25++) {
                                                for (n26 = 0; n26 <= cisl; n26++) {
                                                    for (n33 = 0; n33 <= mudl; n33++) {
                                                        for (n34 = 0; n34 <= cisl; n34++) {
                                                            for (n35 = 0; n35 <= cisl; n35++) {
                                                                for (n36 = 0; n36 <= cisl; n36++) {
                                                                    for (n44 = 0; n44 <= mudl; n44++) {
                                                                        for (n45 = 0; n45 <= cisl; n45++) {
                                                                            for (n46 = 0; n46 <= cisl; n46++) {
                                                                                for (n55 = 0; n55 <= mudl; n55++) {
                                                                                    for (n56 = 0; n56 <= cisl; n56++) {
                                                                                        for (n66 = 0; n66 <= mudl; n66++) {
                                                                                            if (valido(n11, n12, n13, n14, n15, n16, n22, n23, n24, n25,
                                                                                                n26, n33, n34, n35, n36, n44, n45, n46, n55, n56, n66)) {
                                                                                                (*f) << n11 << " " << n12 << " " << n13 << " " << n14 << " "
                                                                                                    << n15 << " " << n16 << " " << n22 << " " << n23 << " "
                                                                                                    << n24 << " " << n25 << " " << n26 << " " << n33 << " "
                                                                                                    << n34 << " " << n35 << " " << n36 << " " << n44 << " "
                                                                                                    << n45 << " " << n46 << " " << n55 << " " << n56 << " "
                                                                                                    << n66 << "\n";
                                                                                            }
                                                                                        }
                                                                                    }
                                                                                }
                                                                            }
                                                                        }
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }
}

f->close();
delete f;
}

//verifica se um estado eh valido
bool valido(int n11, int n12, int n13, int n14, int n15, int n16, int n22,
            int n23, int n24, int n25, int n26, int n33, int n34, int n35,
            int n36, int n44, int n45, int n46, int n55, int n56, int n66)
{
    bool ret = true;
    if (((2 * n11) + n12 + n13 + n14 + n15 + n16) > cudl) {
        ret = false;
    }
    else if ((n12 + (2 * n22) + n23 + n24 + n25 + n26) > cudl) {
        ret = false;
    }
    else if ((n13 + n23 + (2 * n33) + n34 + n35 + n36) > cudl) {
        ret = false;
    }
    else if ((n14 + n24 + n34 + (2 * n44) + n45 + n46) > cudl) {
        ret = false;
    }
    else if ((n15 + n25 + n35 + n45 + (2 * n55) + n56) > cudl) {
        ret = false;
    }
    else if ((n16 + n26 + n36 + n46 + n56 + (2 * n66)) > cudl) {
        ret = false;
    }
    else if ((n12 + n13 + n14 + n25 + n26 + n36) > cisl) {
        ret = false;
    }
    else if ((n13 + n14 + n23 + n24 + n36) > cisl) {
        ret = false;
    }
    else if ((n14 + n24 + n34 + n35) > cisl) {
        ret = false;
    }
    else if ((n35 + n45 + n46) > cisl) {
        ret = false;
    }
    else if ((n15 + n25 + n46 + n56) > cisl) {
        ret = false;
    }
    else if ((n15 + n16 + n25 + n26 + n36) > cisl) {
        ret = false;
    }
    return ret;
}

bool svalido(dtSis* d) {
    return valido(d->v[0], d->v[1], d->v[2], d->v[3], d->v[4],
                d->v[5], d->v[6], d->v[7], d->v[8], d->v[9], d->v[10],
                d->v[11], d->v[12], d->v[13], d->v[14], d->v[15],
                d->v[16], d->v[17], d->v[18], d->v[19], d->v[20]);
}

//le arquivo com os valores de cudl e cisl

```

```

void leConf(void) {
    fstream* f;
    if ((f = new fstream("conf.txt", ios::in)) == 0) {
        cout << "\nErro ao criar objeto";
        return;
    }
    if (!f->is_open()) {
        cout << "\nErro ao abrir arquivo de configuração";
        delete f;
        return;
    }
    *f >> cudl >> cisl;
    criaTabFat(cudl > cisl ? cudl : cisl);
    f->close();
    delete f;
}

//cria uma tabela de valores de fatoriais
void criaTabFat(int n) {
    tabFat = new int[n + 1];
    tabFat[0] = 1;
    int i, f = 1;
    for (i = 1; i <= n; i++) {
        f *= i;
        tabFat[i] = f;
    }
}

//calcula as probabilidades de bloqueio
void proBlo(void) {
    cout << "\nLeitura do arquivo do sistema";
    fstream* fe;
    //i
    fstream* fs;
    fstream* ft;
    //f
    double G;
    int i, tmp, tamT, tamL;
    int perc = -1, percA;
    dtSis dt, tdt;
    unsigned long int coun = 0;
    double probEst;

    for (i = 0; i < 21; i++) {
        proBloSis[i] = 0;
    }

    if ((fe = new fstream("completo.txt", ios::in | ios::binary)) == 0)
    {
        cout << "\nErro ao criar objeto";
        return;
    }
    if (!fe->is_open()) {
        cout << "\nErro ao abrir arquivo";
        return;
    }

    //i
    if ((fs = new fstream("tmp.txt", ios::out)) == 0) {
        cout << "\nErro ao abrir arquivo";
        return;
    }
}

```

```

}
//f

fe->seekg(0, ios::end);
tamT = fe->tellg();
fe->seekg(0);

cout << "\n";
G = 0;
do {
    probEst = 1;

    //le as 21 dimensoes do estado e calcula a probabilidade limite
    for (i = 0; i < 21; i++) {
        *fe >> tmp;

        if (fe->eof()) {
            break;
        }

        dt.v[i] = tmp;

        probEst *= ((powl(chegada[i], tmp)) / (powl(saida[i], tmp) *
                                                    tabFat[tmp]));
    }

    //codigo para mostrar a percentagem do arquivo lida
    tamL = fe->tellg();
    percA = perc;
    perc = (int) (((float) tamL / tamT) * 100);
    if (perc == 0) perc = 100;
    if (perc != percA) {
        cout << (char)13 << perc << " %";
    }

    if (fe->eof()) {
        break;
    }

    //calcula a probabilidade de bloqueio do estado lido
    for (i = 0; i < 21; i++) {
        tdt = dt;
        tdt.v[i]++;
        if (!svalido(&tdt)) {
            proBloSis[i] += probEst;
        }
    }

    //i
    *fs << probEst << "\n";
    //f

    G += probEst;
    coun++;
} while (!fe->eof());

for (i = 0; i < 21; i++) {
    proBloSis[i] /= G;
}

```

```

//i
fs->close();
delete fs;

if (((fs = new fstream("tmp.txt", ios::in)) == 0) || ((ft = new
    fstream("problim.txt", ios::out | ios::trunc) == 0)) {
    cout << "\nErro ao finalizar calculo de probabilidades limite";
    return;
}

if ((!fs->is_open()) || (!ft->is_open())) {
    cout << "\nErro ao finalizar calculo de probabilidades limite";
    return;
}
float ent;

do {
    *fs >> ent;
    *ft << ent / G << "\n";
} while (!fs->eof());
*ft << "\nG: " << G;

ft->close();
fs->close();

delete fs;
delete ft;
//f

cout << "\nCriando arquivo de probabilidades de bloqueio";
if ((fs = new fstream("problo.txt", ios::out)) == 0) {
    cout << "\nErro ao criar objeto";
    return;
}

for (i = 0; i < 21; i++) {
    *fs << proBloSis[i] << "\n";
}

fs->close();
delete fs;

cout << "\nLidos " << coun << " registros";
fe->close();
delete fe;
}

//le o arquivo com as taxas de chegada e saida
void leTaxas(void) {
    fstream* f;
    if ((f = new fstream("taxas.txt", ios::in)) == 0) {
        cout << "\nErro ao criar objeto";
        return;
    }
    if (!f->is_open()) {
        cout << "\nErro ao abrir arquivo de taxas";
        delete f;
        return;
    }
}

```

```

bool chf = true;
int c = 0;
double v;
while (*f >> v) {
    if (chf) {
        chf = false;
        chegada[c] = v;
    }
    else {
        chf = true;
        saida[c] = v;
        c++;
    }
}
delete f;
}

void desempenho(void) {
    fstream* fe;
    fstream* fl;
    fstream* fb;
    int i, nij;
    int tmp, tamT, tamL;
    int perc = -1, percA;
    double lij;
    double probEst, proB;

    //abertura dos arquivos com os estados e as probabilidades limite
    if (((fe = new fstream("completo.txt", ios::in | ios::binary)) == 0) ||
        ((fl = new fstream("problim.txt", ios::in | ios::binary)) == 0)) {
        cout << "\nErro ao criar objeto";
        return;
    }
    if ((!fe->is_open()) || (!fl->is_open())) {
        cout << "\nErro ao abrir os arquivos de entrada";
        delete fe;
        delete fl;
        return;
    }

    //limpeza das variaveis
    for (i = 0; i < 12; i++) {
        medcha[i] = medcan[i] = rejcha[i] = 0;
    }
    for (i = 12; i < 21; i++) {
        medcha[i] = rejcha[i] = 0;
    }

    //leitura dos estados e probabilidades limite
    //calculo do numero medio de chamada entre os satelites
    fe->seekg(0, ios::end);
    tamT = fe->tellg();
    fe->seekg(0);

    do {
        probEst = 1;

        //le as 21 dimensoes do estado
        *fl >> lij;
        for (i = 0; i < 21; i++) {

```

```

        *fe >> nij;

        if (fe->eof()) {
            break;
        }

        medcha[i] += nij * lij;
    }

    //codigo para mostrar a percentagem do arquivo lida
    tamL = fe->tellg();
    percA = perc;
    perc = (int) (((float) tamL / tamT) * 100);
    if (perc == 0) perc = 100;
    if (perc != percA) {
        cout << (char)13 << perc << " %";
    }

} while (!fe->eof());

fe->close();
fl->close();

delete fe;
delete fl;

//calculo do numero de canais medio utilizados em cada conexao fisica
medcan[0] = (2 * medcha[0]) + medcha[1] + medcha[2] + medcha[3] +
            medcha[4] + medcha[5];
medcan[1] = (2 * medcha[6]) + medcha[1] + medcha[7] + medcha[8] +
            medcha[9] + medcha[10];
medcan[2] = (2 * medcha[11]) + medcha[2] + medcha[7] + medcha[12] +
            medcha[13] + medcha[14];
medcan[3] = (2 * medcha[15]) + medcha[3] + medcha[8] + medcha[12] +
            medcha[16] + medcha[17];
medcan[4] = (2 * medcha[18]) + medcha[4] + medcha[9] + medcha[13] +
            medcha[16] + medcha[19];
medcan[5] = (2 * medcha[20]) + medcha[5] + medcha[10] + medcha[14] +
            medcha[17] + medcha[19];
medcan[6] = medcha[1] + medcha[2] + medcha[3] + medcha[9] +
            medcha[10] + medcha[14];
medcan[7] = medcha[2] + medcha[3] + medcha[7] + medcha[8] + medcha[14];
medcan[8] = medcha[3] + medcha[8] + medcha[12] + medcha[13];
medcan[9] = medcha[13] + medcha[16] + medcha[17];
medcan[10] = medcha[4] + medcha[9] + medcha[17] + medcha[19];
medcan[11] = medcha[4] + medcha[5] + medcha[9] + medcha[10] +
            medcha[14];

//calculo das taxas de rejeicao de chamadas
//total de rejeicao de chamadas e taxa total de chegada
if ((fb = new fstream("problo.txt", ios::in | ios::binary)) == 0) {
    cout << "\nErro ao criar obejto";
    return;
}
if (!fb->is_open()) {
    cout << "\nErro ao abrir arquivo de entrada";
    delete fb;
    return;
}

cheTotal = rejTotal = 0;

```

```

i = 0;
while (*fb >> proB) {
    rejTotal += (rejcha[i] = (proB * chegada[i]));
    cheTotal += chegada[i];
    i++;
}

fb->close();
delete fb;

//probabilidade de bloqueio do sistema
proBSis = rejTotal / cheTotal;

//taxas de atendimento
for (i = 0; i < 21; i++) {
    taxaAtend[i] = chegada[i] - rejcha[i];
}

//salva os dados
if ((fe = new fstream("desempenho.txt", ios::out | ios::trunc)) == 0)
{
    cout << "\nErro ao abrir arquivo de saída";
    return;
}

*fe << "Número médio de chamadas entre os satelites\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << medcha[i];
}
*fe << "\n\nNúmero médio de canais em conexão física\n";
for (i = 0; i < 12; i++) {
    *fe << "\n" << medcan[i];
}
*fe << "\n\nTaxa de rejeição de chamadas\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << rejcha[i];
}
*fe << "\n\nTaxa total de rejeicao do sistema: " << rejTotal;
*fe << "\n\nTaxa total de chegada do sistema: " << cheTotal;
*fe << "\n\nProbabilidade de bloqueio do sistema: " << proBSis;
*fe << "\n\nTaxa de chegada efetiva\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << taxaAtend[i];
}
fe->close();
delete fe;
cout << "\nPronto";
}

```

A.2 Código da implementação do modelo markoviano do sistema de seis satélites não-geostacionários (com *handoff*) apresentado no Capítulo 4

```
#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

struct dtSis {
    int v[21];
};

//declaracao das funcoes
void gera(void);
void leConf(void);
void criaTabFat(int n);
bool valido(int, int, int,
            int, int, int, int, int, int, int, int);
void proBlo(void);
bool svalido(dtSis*);
void leTaxas(void);
void novastaxas(dtSis);
void desempenho(void);
void proBloEst(double, dtSis);
dtSis leEstado(fstream*);
void desempenho_QCH(void);
bool verifica_handoff(dtSis, int);
int verifica_conexao(int);

//declaracao das variaveis globais
int* tabFat = 0;
int cudl, cisl;
double alfa;

//vetores com as taxas de chegada e servico do sistema completo
double chegada[21], servico[21];
//vetores com as novas taxas de chegada e servico para cada estado lido
double nchegada[21], nservico[21];
double proBloSis[21];

//medidas de desempenho
double medcha[21], medcan[12], rejcha[21], taxaAtend[21];
double rejTotal, proBSis, cheTotal;
double medQCH[21];

int main()
{
    int op;
    leConf();
    leTaxas();
    do {
        cout << "\nSelecione\n1 - Gerar estados do sistema";
        cout << "\n2 - Calcular probabilidades de bloqueio";
        cout << "\n3 - Calcular medidas de desempenho";
        cout << "\n4 - Calcular medida de desempenho QCH";
        cout << "\n5 - Sair\n";
        cin >> op;
        switch (op) {
            case 1:
```

```

        gera();
        break;
    case 2:
        proBlo();
        break;
    case 3:
        desempenho();
        break;
    case 4:
        desempenho_QCH();
        break;
    }
} while (op != 5);
if (tabFat != 0) {
    delete tabFat;
}
return 0;
}

//gera os estados do sistema
void gera(void) {
    fstream* f = NULL;
    if ((f = new fstream("completo.txt", ios::out | ios::trunc)) == 0) {
        cout << "\nErro ao abrir arquivo";
        return;
    }
    int n11, n12, n13, n14, n15, n16, n22, n23, n24, n25, n26, n33, n34,
        n35, n36, n44, n45, n46, n55, n56, n66;
    int mudl;
    mudl = cudl / 2;
    for (n11 = 0; n11 <= mudl; n11++) {
        for (n12 = 0; n12 <= cisl; n12++) {
            for (n13 = 0; n13 <= cisl; n13++) {
                for (n14 = 0; n14 <= cisl; n14++) {
                    for (n15 = 0; n15 <= cisl; n15++) {
                        for (n16 = 0; n16 <= cisl; n16++) {
                            for (n22 = 0; n22 <= mudl; n22++) {
                                for (n23 = 0; n23 <= cisl; n23++) {
                                    for (n24 = 0; n24 <= cisl; n24++) {
                                        for (n25 = 0; n25 <= cisl; n25++) {
                                            for (n26 = 0; n26 <= cisl; n26++) {
                                                for (n33 = 0; n33 <= mudl; n33++) {
                                                    for (n34 = 0; n34 <= cisl; n34++) {
                                                        for (n35 = 0; n35 <= cisl; n35++) {
                                                            for (n36 = 0; n36 <= cisl; n36++) {
                                                                for (n44 = 0; n44 <= mudl; n44++) {
                                                                    for (n45 = 0; n45 <= cisl; n45++) {
                                                                        for (n46 = 0; n46 <= cisl; n46++) {
                                                                            for (n55 = 0; n55 <= mudl; n55++) {
                                                                                for (n56 = 0; n56 <= cisl; n56++) {
                                                                                    for (n66 = 0; n66 <= mudl; n66++) {
                                                                                        if (valido(n11, n12, n13, n14, n15, n16, n22, n23, n24, n25, n26,
                                                                                            n33, n34, n35, n36, n44, n45, n46, n55, n56, n66)) {
                                                                                            (*f) << n11 << " " << n12 << " " << n13 << " " << n14 << " "
                                                                                                << n15 << " " << n16 << " " << n22 << " " << n23 << " "
                                                                                                << n24 << " " << n25 << " " << n26 << " " << n33 << " "
                                                                                                << n34 << " " << n35 << " " << n36 << " " << n44 << " "
                                                                                                << n45 << " " << n46 << " " << n55 << " " << n56 << " "
                                                                                                << n66 << "\n";
                                                                                        }
                                                                                    }
                                                                                }
                                                                            }
                                                                        }
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
}

```



```

        ret = false;
    }
    else if ((n15 + n16 + n25 + n26 + n36) > cisl) {
        ret = false;
    }
    return ret;
}

bool svalido(dtSis* d) {
    return valido(d->v[0], d->v[1], d->v[2], d->v[3], d->v[4],
                 d->v[5], d->v[6], d->v[7], d->v[8], d->v[9], d->v[10],
                 d->v[11], d->v[12], d->v[13], d->v[14], d->v[15],
                 d->v[16], d->v[17], d->v[18], d->v[19], d->v[20]);
}

//le arquivo com os valores de cudl, cisl e alfa
void leConf(void) {
    fstream* f;
    if ((f = new fstream("conf.txt", ios::in)) == 0) {
        cout << "\nErro ao criar objeto";
        return;
    }
    if (!f->is_open()) {
        cout << "\nErro ao abrir arquivo de configuração";
        delete f;
        return;
    }
    *f >> cudl >> cisl;
    *f >> alfa;
    criaTabFat(cudl > cisl ? cudl : cisl);
    f->close();
    delete f;
}

//cria uma tabela de valores de fatoriais
void criaTabFat(int n) {
    tabFat = new int[n + 1];
    tabFat[0] = 1;
    int i, f = 1;
    for (i = 1; i <= n; i++) {
        f *= i;
        tabFat[i] = f;
    }
}

//calcula as probabilidades de bloqueio
void probBlo(void) {
    cout << "\nCalculando probabilidades de bloqueio";
    cout << "\nLeitura do arquivo do sistema";
    fstream* fe;
    //i
    fstream* fs;
    fstream* ft;
    //f
    double G;
    int i, tamT, tamL;
    int perc = -1, percA;
    dtSis dt;
    unsigned long int coun = 0;
    double probEst;
}

```

```

for (i = 0; i < 21; i++) {
    proBloSis[i] = 0;
}

if ((fe = new fstream("completo.txt", ios::in | ios::binary)) == 0)
{
    cout << "\nErro ao criar objeto";
    return;
}
if (!fe->is_open()) {
    cout << "\nErro ao abrir arquivo";
    return;
}

//i
if ((fs = new fstream("tmp.txt", ios::out)) == 0) {
    cout << "\nErro ao criar objeto";
    return;
}

//if ()
//f

fe->seekg(0, ios::end);
tamT = fe->tellg();
fe->seekg(0);

cout << "\n";
G = 0;
do {

    //le as 21 dimensoes do estado e calcula a probabilidade limite
    dt = leEstado(fe);

    //codigo para mostrar a percentagem do arquivo lida
    tamL = fe->tellg();
    percA = perc;
    perc = (int) (((float) tamL / tamT) * 100);
    if (perc == 0) perc = 100;
    if (perc != percA) {
        cout << (char)13 << perc << " %";
    }

    if (fe->eof()) {
        break;
    }

    //calcular novas taxas de chegada e de servico
    novastaxas(dt);

    probEst = 1;
    //calcula a probabilidade limite do estado
    for (i = 0; i < 21; i++)
    {
        probEst *= ((powl(nchegada[i], dt.v[i])) /
                    (powl(nservico[i], dt.v[i]) * tabFat[dt.v[i]]));
    }

    //calcula a probabilidade de bloqueio do estado lido

```

```

        probBloEst(probEst, dt);

        //i
        *fs << probEst << "\n";
        //f

        G += probEst;
        count++;
    } while (!fe->eof());

    for (i = 0; i < 21; i++) {
        probBloSis[i] /= G;
    }

    //i
    fs->close();
    delete fs;

    if (((fs = new fstream("tmp.txt", ios::in)) == 0) ||
        ((ft = new fstream("problim.txt", ios::out | ios::trunc)) == 0))
    {
        cout << "\nErro ao finalizar calculo de probabilidades limite";
        return;
    }

    if ((!fs->is_open()) || (!ft->is_open())) {
        cout << "\nErro ao finalizar calculo de probabilidades limite";
        return;
    }

    float ent;

    cout << "\nCriando arquivo de probabilidades limite\n";

    do {
        *fs >> ent;

        if (fs->eof()) {
            break;
        }

        *ft << ent / G << "\n";
    } while (!fs->eof());
    *ft << "\nG: " << G;

    ft->close();
    fs->close();

    delete fs;
    delete ft;
    //f

    cout << "\nCriando arquivo de probabilidades de bloqueio";
    if ((fs = new fstream("problo.txt", ios::out)) == 0) {
        cout << "\nErro ao criar objeto";
        return;
    }

    for (i = 0; i < 21; i++) {

```

```

        *fs << proBloSis[i] << "\n";
    }

    fs->close();
    delete fs;

    cout << "\nLidos " << coun << " registros\n";
    fe->close();
    delete fe;

}

//le o arquivo com a taxa de chegada em cada conexao e a taxa de servico
void leTaxas(void) {
    fstream* f;
    if ((f = new fstream("taxas.txt", ios::in)) == 0) {
        cout << "\nErro ao criar objeto";
        return;
    }
    if (!f->is_open()) {
        cout << "\nErro ao abrir arquivo de taxas";
        delete f;
        return;
    }
    bool chf = true;
    int c = 0;
    double v;
    while (*f >> v) {
        if (chf) {
            chf = false;
            chegada[c] = v;
        }
        else {
            chf = true;
            servico[c] = v;
            c++;
        }
    }
    delete f;
}

void novastaxas(dtSis dt) {
    nchegada[0] = chegada[0] + (alfa * dt.v[5]);
    nchegada[1] = chegada[1] + (alfa * dt.v[10]) + (2 * alfa * dt.v[0]);
    nchegada[2] = chegada[2] + (alfa * dt.v[1]) + (alfa * dt.v[14]);
    nchegada[3] = chegada[3] + (alfa * dt.v[2]) + (alfa * dt.v[17]);
    nchegada[4] = chegada[4] + (alfa * dt.v[3]) + (alfa * dt.v[19]);
    nchegada[5] = chegada[5] + (alfa * dt.v[4]) + (2 * alfa * dt.v[20]);
    nchegada[6] = chegada[6] + (alfa * dt.v[1]);
    nchegada[7] = chegada[7] + (alfa * dt.v[2]) + (2 * alfa * dt.v[6]);
    nchegada[8] = chegada[8] + (alfa * dt.v[3]) + (alfa * dt.v[7]);
    nchegada[9] = chegada[9] + (alfa * dt.v[4]) + (alfa * dt.v[8]);
    nchegada[10] = chegada[10] + (alfa * dt.v[5]) + (alfa * dt.v[9]);
    nchegada[11] = chegada[11] + (alfa * dt.v[7]);
    nchegada[12] = chegada[12] + (alfa * dt.v[8]) + (2*alfa * dt.v[11]);
    nchegada[13] = chegada[13] + (alfa * dt.v[9]) + (alfa * dt.v[12]);
    nchegada[14] = chegada[14] + (alfa * dt.v[10]) + (alfa * dt.v[13]);
    nchegada[15] = chegada[15] + (alfa * dt.v[12]);
    nchegada[16] = chegada[16] + (alfa * dt.v[13]) + (2*alfa * dt.v[15]);
    nchegada[17] = chegada[17] + (alfa * dt.v[14]) + (alfa * dt.v[16]);
}

```

```

nchegada[18] = chegada[18] + (alfa * dt.v[16]);
nchegada[19] = chegada[19] + (alfa * dt.v[17]) + (2*alfa * dt.v[18]);
nchegada[20] = chegada[20] + (alfa * dt.v[19]);

nservico[0] = servico[0] + (2 * alfa * dt.v[0]);
nservico[1] = servico[1] + (2 * alfa * dt.v[1]);
nservico[2] = servico[2] + (2 * alfa * dt.v[2]);
nservico[3] = servico[3] + (2 * alfa * dt.v[3]);
nservico[4] = servico[4] + (2 * alfa * dt.v[4]);
nservico[5] = servico[5] + (2 * alfa * dt.v[5]);
nservico[6] = servico[6] + (2 * alfa * dt.v[6]);
nservico[7] = servico[7] + (2 * alfa * dt.v[7]);
nservico[8] = servico[8] + (2 * alfa * dt.v[8]);
nservico[9] = servico[9] + (2 * alfa * dt.v[9]);
nservico[10] = servico[10] + (2 * alfa * dt.v[10]);
nservico[11] = servico[11] + (2 * alfa * dt.v[11]);
nservico[12] = servico[12] + (2 * alfa * dt.v[12]);
nservico[13] = servico[13] + (2 * alfa * dt.v[13]);
nservico[14] = servico[14] + (2 * alfa * dt.v[14]);
nservico[15] = servico[15] + (2 * alfa * dt.v[15]);
nservico[16] = servico[16] + (2 * alfa * dt.v[16]);
nservico[17] = servico[17] + (2 * alfa * dt.v[17]);
nservico[18] = servico[18] + (2 * alfa * dt.v[18]);
nservico[19] = servico[19] + (2 * alfa * dt.v[19]);
nservico[20] = servico[20] + (2 * alfa * dt.v[20]);
}

void desempenho(void) {
    fstream* fe;
    fstream* fl;
    fstream* fb;
    int i, nij;
    int tamT, tamL;
    int perc = -1, percA;
    double lij;
    double probEst, probB;

    cout << "\nCalculando medidas de desempenho\n";

    //abertura dos arquivos com os estados e as probabilidades limite
    if (((fe = new fstream("completo.txt", ios::in | ios::binary)) == 0) ||
        ((fl = new fstream("problim.txt", ios::in | ios::binary)) == 0))
    {
        cout << "\nErro ao criar objeto";
        return;
    }
    if ((!fe->is_open()) || (!fl->is_open())) {
        cout << "\nErro ao abrir os arquivos de entrada";
        delete fe;
        delete fl;
        return;
    }

    //limpeza das variaveis
    for (i = 0; i < 12; i++) {
        medcha[i] = medcan[i] = rejcha[i] = 0;
    }
    for (i = 12; i < 21; i++) {
        medcha[i] = rejcha[i] = 0;
    }
}

```

```

//leitura dos estados e probabilidades limite
//calculo do numero medio de chamada entre os satelites
fe->seekg(0, ios::end);
tamT = fe->tellg();
fe->seekg(0);

do {
    probEst = 1;

    *fl >> lij;
    //le as 21 dimensoes do estado
    for (i = 0; i < 21; i++) {
        *fe >> nij;

        if (fe->eof()) {
            break;
        }
        medcha[i] += nij * lij;
    }

    //codigo para mostrar a percentagem do arquivo lida
    tamL = fe->tellg();
    percA = perc;
    perc = (int) (((float) tamL / tamT) * 100);
    if (perc == 0) perc = 100;
    if (perc != percA) {
        cout << (char)13 << perc << " %";
    }
} while (!fe->eof());

fe->close();
fl->close();

delete fe;
delete fl;

//calculo do numero de canais medio utilizados em cada conexao fisica
medcan[0] = (2 * medcha[0]) + medcha[1] + medcha[2] + medcha[3] +
            medcha[4] + medcha[5];
medcan[1] = (2 * medcha[6]) + medcha[1] + medcha[7] + medcha[8] +
            medcha[9] + medcha[10];
medcan[2] = (2 * medcha[11]) + medcha[2] + medcha[7] + medcha[12] +
            medcha[13] + medcha[14];
medcan[3] = (2 * medcha[15]) + medcha[3] + medcha[8] + medcha[12] +
            medcha[16] + medcha[17];
medcan[4] = (2 * medcha[18]) + medcha[4] + medcha[9] + medcha[13] +
            medcha[16] + medcha[19];
medcan[5] = (2 * medcha[20]) + medcha[5] + medcha[10] + medcha[14] +
            medcha[17] + medcha[19];

medcan[6] = medcha[1] + medcha[2] + medcha[3] + medcha[9] +
            medcha[10] + medcha[14];
medcan[7] = medcha[2] + medcha[3] + medcha[7] + medcha[8] + medcha[14];
medcan[8] = medcha[3] + medcha[8] + medcha[12] + medcha[13];
medcan[9] = medcha[13] + medcha[16] + medcha[17];
medcan[10] = medcha[4] + medcha[9] + medcha[17] + medcha[19];
medcan[11] = medcha[4] + medcha[5] + medcha[9] + medcha[10] +
            medcha[14];

```

```

//calculo das taxas de rejeicao de chamadas
//total de rejeicao de chamadas e taxa total de chegada
if ((fb = new fstream("problo.txt", ios::in | ios::binary)) == 0) {
    cout << "\nErro ao criar obejto";
    return;
}
if (!fb->is_open()) {
    cout << "\nErro ao abrir arquivo de entrada";
    delete fb;
    return;
}

cheTotal = rejTotal = 0;
i = 0;
while (*fb >> proB) {
    rejTotal += (rejcha[i] = (proB * chegada[i]));
    cheTotal += chegada[i];
    i++;
}

fb->close();
delete fb;

//probabilidade de bloqueio do sistema
proBSis = rejTotal / cheTotal;

//taxas de atendimento
for (i = 0; i < 21; i++) {
    taxaAtend[i] = chegada[i] - rejcha[i];
}

//salva os dados
if ((fe = new fstream("desempenho.txt", ios::out | ios::trunc)) == 0)
{
    cout << "\nErro ao abrir arquivo de saída";
    return;
}

*fe << "Número médio de chamadas entre os satelites\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << medcha[i];
}
*fe << "\n\nNúmero médio de canais em conexão física\n";
for (i = 0; i < 12; i++) {
    *fe << "\n" << medcan[i];
}
*fe << "\n\nTaxa de rejeição de chamadas\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << rejcha[i];
}
*fe << "\n\nTaxa total de rejeicao do sistema: " << rejTotal;
*fe << "\n\nTaxa total de chegada do sistema: " << cheTotal;
*fe << "\n\nProbabilidade de bloqueio do sistema: " << proBSis;
*fe << "\n\nTaxa de chegada efetiva\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << taxaAtend[i];
}
fe->close();
delete fe;
cout << "\nPronto\n";
}

```

```

void probBloEst(double probEst, dtSis dt) {
    int i;
    dtSis tdt;
    tdt = dt;
    for (i = 0; i < 21; i++) {
        tdt = dt;
        tdt.v[i]++;
        if (!svalido(&tdt)) {
            probBloSis[i] += probEst;
        }
    }
}

dtSis leEstado(fstream* fe) {
    int tmp, i;
    dtSis dt;
    for (i = 0; i < 21; i++) {
        *fe >> tmp;
        if (fe->eof()) {
            break;
        }
        dt.v[i] = tmp;
    }
    return dt;
}

void desempenho_QCH(void) {
    fstream* fe;
    fstream* fl;
    dtSis est;
    int i, fm;
    double lij;
    int tamT, tamL;
    int perc = -1, percA;

    cout << "\nCalculando medida de desempenho QCH\n";

    //abertura dos arquivos com os estados e as probabilidades limite
    if (((fe = new fstream("completo.txt", ios::in | ios::binary)) == 0) ||
        ((fl = new fstream("problim.txt", ios::in | ios::binary)) == 0))
    {
        cout << "\nErro ao criar objeto";
        return;
    }
    if ((!fe->is_open()) || (!fl->is_open())) {
        cout << "\nErro ao abrir os arquivos de entrada";
        delete fe;
        delete fl;
        return;
    }

    //limpeza das variaveis
    for (i = 0; i < 21; i++) {
        medQCH[i] = 0;
    }

    fe->seekg(0, ios::end);
    tamT = fe->tellg();
    fe->seekg(0);

    //itera sobre todos os estados do sistema

```

```

do {
    //le as 21 dimensoes do estado
    est = leEstado(fe);

    //codigo para mostrar a percentagem do arquivo lida
    tamL = fe->tellg();
    percA = perc;
    perc = (int) (((float) tamL / tamT) * 100);
    if (perc == 0) perc = 100;
    if (perc != percA) {
        cout << (char)13 << perc << " %";
    }

    if (fe->eof()) {
        break;
    }
    //le a probabilidade limite do estado
    *fl >> lij;
    //itera sobre todas as conexões
    for (i = 0; i < 21; i++) {
        if (verifica_handoff(est, i)) {
            //verifica se na conexao Ni-j i == j (fm = 2) ou i != j (fm = 1)
            fm = verifica_conexao(i);
            //incrementa valor da medida de desempenho da conexão
            medQCH[i] += /*fm */ alfa * est.v[i] * lij;
        }
    }
} while (!fe->eof());
fe->close();
fl->close();

//escreve os valores no arquivo
if ((fe = new fstream("desempenho_QCH.txt", ios::out | ios::trunc)) ==
0)
{
    cout << "\nErro ao abrir arquivo de saída";
    return;
}

*fe << "Quantidade de chamadas em cada conexão devido ao HANDOFF\n";
for (i = 0; i < 21; i++) {
    *fe << "\n" << medQCH[i];
}

fe->close();
cout << "\nPronto\n";
}

bool verifica_handoff(dtSis est, int con) {
    dtSis tmp_est, cmp_est;
    int j;
    tmp_est = est;
    if (tmp_est.v[con] == 0) {
        //nenhuma chamada na conexao selecionada, sem handoff
        return false;
    }
    //decrementa o valor na conexão selecionada
    tmp_est.v[con]--;
    //itera sobre todas as conexões, exceto a selecionada
    for (j = 0; j < 21; j++) {
        if (j != con) {

```

```

        //incrementa o valor da conexao j na estrutura de comparação
        cmp_est = tmp_est;
        cmp_est.v[j]++;
        //verifica se o estado obtido é válido
        if (svalido(&cmp_est)) {
            //se for então o estado tem handoff
            return true;
        }
    }
}
//nao encontrou nenhum estado, sem handoff
return false;
}

int verifica_conexao(int i) {
    int ret = 1;
    int vet_iguais[] = {0, 6, 11, 15, 18, 20};
    int j;
    for (j = 0; j < 6; j++) {
        if (vet_iguais[j] == i) {
            ret = 2;
            break;
        }
    }
    return ret;
}
}

```

A.3 Código da implementação do modelo markoviano do sistema de seis satélites geostacionários com decomposição, utilizando as equações de probabilidade de bloqueio entre subsistemas sugeridas por Zaim (2001); Zaim et al. (2003); e sugeridas por este trabalho, apresentadas no Capítulo 5

```
#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

struct dtSub {
    int v[12];
};

//declaracao das funcoes
void gSub1(void);
bool vSub1(int, int, int);
bool vsSub1(dtSub*);
void calculaGSub1(void);
void proBloSub1(void);
void taxaSub1(void);
void gSub2(void);
bool vSub2(int, int, int);
bool vsSub2(dtSub*);
void taxaSub2(void);
void calculaGSub2(void);
void leConf(void);
void leTaxas(void);
void criaTabFat(int n);
void proBloqueio(void);
void taxaSub2(void);
double probEst(dtSub*);
bool converge(void);
void salva(void);

//declaracao das variaveis globais
//vetores com as taxas de chegada e saida do sistema completo
double chegada[21], saida[21];
//vetores com as taxas de chegada e saida do sub-sistema 1
double cheSub1[12], saiSub1[12];
//vetores com as taxas de chegada e saida do sub-sistema 2
double cheSub2[12], saiSub2[12];
//vetores com as probabilidades de bloqueio dos sistemas
double blSub1[12], blSub2[12];
//vetores com as probabilidades de bloqueio
double vetBlo1[9], vetBlo2[9];

int cudl, cisl;
int* tabFat = 0;
double GS1 = 0, GS2 = 0;

int main() {
    int op;
    leConf();
    do {
```



```

        }
    }
}

f->close();
delete f;
}

//verifica se um determinado estado eh valido no subsistema 1
bool vSub1(int n11, int n12, int n13, int n1n1, int n1s1, int n22,
           int n23, int n2n1, int n2s1, int n33, int n3n1, int n3s1)
{
    bool ret = true;
    if (((2 * n11) + n12 + n13 + n1n1 + n1s1) > cud1) {
        ret = false;
    }
    else if ((n12 + (2 * n22) + n23 + n2n1 + n2s1) > cud1) {
        ret = false;
    }
    else if ((n13 + n23 + (2 * n33) + n3n1 + n3s1) > cud1) {
        ret = false;
    }
    else if ((n1n1 + n2n1 + n3n1) > cis1) {
        ret = false;
    }
    else if ((n12 + n13 + n1s1 + n2n1 + n3n1) > cis1) {
        ret = false;
    }
    else if ((n13 + n23 + n1s1 + n2s1 + n3n1) > cis1) {
        ret = false;
    }
    else if ((n1s1 + n2s1 + n3s1) > cis1) {
        ret = false;
    }
    return ret;
}

bool vsSub1(dtSub* dt) {
    return vSub1(dt->v[0], dt->v[1], dt->v[2], dt->v[3], dt->v[4],
                dt->v[5], dt->v[6], dt->v[7], dt->v[8], dt->v[9],
                dt->v[10], dt->v[11]);
}

//calcula as taxas para os satelites ficticios do sub-sistema 1
void taxaSub1(void) {
    cheSub1[3] = ((1 - vetBlo2[2]) * chegada[5]) + ((1 - vetBlo2[1]) *
                                                    chegada[4]);

    cheSub1[4] = ((1 - vetBlo2[0]) * chegada[3]);
    cheSub1[7] = ((1 - vetBlo2[5]) * chegada[10]) + ((1 - vetBlo2[4]) *
                                                    chegada[9]);

    cheSub1[8] = ((1 - vetBlo2[3]) * chegada[8]);
    cheSub1[10] = ((1 - vetBlo2[8]) * chegada[14]);
    cheSub1[11] = ((1 - vetBlo2[6]) * chegada[12]) + ((1 - vetBlo2[7]) *
                                                    chegada[13]);

    saiSub1[3] = (((1 - vetBlo2[2]) * chegada[5] * saida[5]) +
                 ((1 - vetBlo2[1]) * chegada[4] * saida[4]))/cheSub1[3];
    saiSub1[4] = saida[3];
    saiSub1[7] = (((1 - vetBlo2[5]) * chegada[10] * saida[10]) +
                 ((1 - vetBlo2[4]) * chegada[9] * saida[9]))/cheSub1[7];
    saiSub1[8] = saida[8];
}

```

```

    saiSub1[10] = saida[14];
    saiSub1[11] = (((1 - vetBlo2[6]) * chegada[12] * saida[12]) +
        ((1 - vetBlo2[7]) * chegada[13] * saida[13]))/cheSub1[11];
}

//calcula G e probabilidades de bloqueio do sub-sistema 1
void calculaGSub1(void) {
    cout << "\nLeitura do arquivo do sistema 1";
    fstream* fe;
    //i
    fstream* fs;
    fstream* ft;
    //f
    dtSub dt, tdt;
    int i, tmp, tamT, tamL;
    int perc = -1, percA;
    unsigned long int coun = 0;
    double probEst;

    if ((fe = new fstream("sistema1.txt", ios::in | ios::binary)) == 0)
    {
        cout << "\nErro ao abrir o arquivo do sistema 1";
        return;
    }

    //i
    if ((fs = new fstream("tmp1.txt", ios::out)) == 0) {
        cout << "\nErro ao abrir arquivo";
        return;
    }
    //f

    fe->seekg(0, ios::end);
    tamT = fe->tellg();
    fe->seekg(0, ios::beg);

    cout << "\n";
    GS1 = 0;
    do {
        probEst = 1;

        //le as 12 dimensoes do estado e calcula a probabilidade limite
        for (i = 0; i < 12; i++) {
            *fe >> tmp;
            if (fe->eof()) {
                break;
            }
            dt.v[i] = tmp;

            probEst *= ((powl(cheSub1[i], tmp)) /
                (powl(saiSub1[i], tmp) * tabFat[tmp]));
        }

        //codigo para mostrar a percentagem do arquivo lida
        tamL = fe->tellg();
        percA = perc;
        perc = (int) (((float) tamL / tamT) * 100);
        if (perc == 0) perc = 100;
        if (perc != percA) {
            cout << (char)13 << perc << " %";
        }
    }
}

```

```

    }

    if (fe->eof()) {
        break;
    }

    //calcula a probabilidade de bloqueio do estado lido
    for (i = 0; i < 12; i++) {
        tdt = dt;
        tdt.v[i]++;
        if (!vsSub1(&tdt)) {
            blSub1[i] += probEst;
        }
    }

    //i
    *fs << probEst << "\n";
    //f

    GS1 += probEst;
    coun++;
} while (!fe->eof());

for (i = 0; i < 12; i++) {
    blSub1[i] /= GS1;
}

//i
fs->close();
delete fs;

if (((fs = new fstream("tmp1.txt", ios::in | ios::binary)) == 0) ||
    ((ft = new fstream("probLimSub1.txt", ios::out | ios::trunc)) == 0))
{
    cout << "\nErro ao finalizar calculo de probabilidades limite -";
    cout << " Sistema 1";
    return;
}
float ent;

cout << "\nCriando arquivo de probabilidades limite";

while (*fs >> ent) {
    *ft << ent / GS1 << "\n";
}

*ft << "\nG: " << GS1;

ft->close();
fs->close();

if (fs != 0) {
    delete fs;
}
if (ft != 0) {
    delete ft;
}
//f

cout << "\nLidos " << coun << " registros";
fe->close();

```

```

    delete fe;
}

void gSub2(void) {
    fstream* f = NULL;
    if ((f = new fstream("sistema2.txt", ios::out | ios::trunc)) == 0) {
        cout << "\nErro ao abrir arquivo";
        return;
    }
    int n44, n45, n46, nn24, ns24, n55, n56, nn25, ns25, n66, nn26, ns26;
    int mudl;
    mudl = cudl / 2;
    ns26 = 0;
    nn24 = 0;
    for (n44 = 0; n44 <= mudl; n44++) {
        for (n45 = 0; n45 <= cisl; n45++) {
            for (n46 = 0; n46 <= cisl; n46++) {
                for (ns24 = 0; ns24 <= cisl; ns24++) {
                    for (n55 = 0; n55 <= mudl; n55++) {
                        for (n56 = 0; n56 <= cisl; n56++) {
                            for (nn25 = 0; nn25 <= cisl; nn25++) {
                                for (ns25 = 0; ns25 <= cisl; ns25++) {
                                    for (n66 = 0; n66 <= mudl; n66++) {
                                        for (nn26 = 0; nn26 <= cisl; nn26++) {
                                            if (vSub2(n44, n45, n46, nn24, ns24, n55, n56, nn25, ns25, n66,
                                                nn26, ns26)) {
                                                (*f) << n44 << " " << n45 << " " << n46 << " " << nn24 << " "
                                                    << ns24 << " " << n55 << " " << n56 << " " << nn25 << " "
                                                    << ns25 << " " << n66 << " " << nn26 << " " << ns26 << "\n";
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    f->close();
    delete f;
}

```

```

//verifica se um determinado estado eh valido no subsistema 2
bool vSub2(int n44, int n45, int n46, int nn24, int ns24, int n55,
    int n56, int nn25, int ns25, int n66, int nn26, int ns26)
{
    bool ret = true;
    if (((2 * n44) + n45 + n46 + nn24 + ns24) > cudl) {
        ret = false;
    }
    else if ((n45 + (2 * n55) + n56 + nn25 + ns25) > cudl) {
        ret = false;
    }
    else if ((n46 + n56 + (2 * n66) + nn26 + ns26) > cudl) {
        ret = false;
    }
    else if ((ns24 + ns25) > cisl) {
        ret = false;
    }
}

```

```

else if ((n45 + n46 + ns25) > cis1) {
    ret = false;
}
else if ((n46 + n56 + nn25) > cis1) {
    ret = false;
}
else if ((nn26 + nn25) > cis1) {
    ret = false;
}
return ret;
}

bool vsSub2(dtSub* dt) {
    return vSub2(dt->v[0], dt->v[1], dt->v[2], dt->v[3], dt->v[4],
                dt->v[5], dt->v[6], dt->v[7], dt->v[8], dt->v[9],
                dt->v[10], dt->v[11]);
}
//calcula as taxas para os satelites ficticios do sub-sistema 2
void taxaSub2(void) {
    cheSub2[3] = 0;
    cheSub2[4] = ((1 - vetBlo1[0]) * chegada[3]) + ((1 - vetBlo1[3]) *
                chegada[8]) + ((1 - vetBlo1[6]) * chegada[12]);
    cheSub2[7] = ((1 - vetBlo1[1]) * chegada[4]) + ((1 - vetBlo1[4]) *
                chegada[9]);
    cheSub2[8] = ((1 - vetBlo1[7]) * chegada[13]);
    cheSub2[10] = ((1 - vetBlo1[2]) * chegada[5]) + ((1 - vetBlo1[5]) *
                chegada[10]) + ((1 - vetBlo1[8]) * chegada[14]);
    cheSub2[11] = 0;

    saiSub2[3] = 0;
    saiSub2[4] = (((1 - vetBlo1[0]) * chegada[3] * saida[3]) +
                ((1 - vetBlo1[3]) * chegada[8] * saida[8]) +
                ((1 - vetBlo1[6]) * chegada[12] * saida[12]))/cheSub2[4];
    saiSub2[7] = (((1 - vetBlo1[1]) * chegada[4] * saida[4]) +
                ((1 - vetBlo1[4]) * chegada[9] * saida[9]))/cheSub2[7];
    saiSub2[8] = saida[13];
    saiSub2[10] = (((1 - vetBlo1[2]) * chegada[5] * saida[5]) +
                ((1 - vetBlo1[5]) * chegada[10] * saida[10]) +
                ((1 - vetBlo1[8]) * chegada[14] * saida[14]))/cheSub2[10];
    saiSub2[11] = 0;
}

//calcula G e probabilidades de bloqueio do sub-sistema 2
void calculaGSub2(void) {
    cout << "\nLeitura do arquivo do sistema 2";
    fstream* fe;
    //i
    fstream* fs;
    fstream* ft;
    //f
    int i, tmp, tamT, tamL;
    int perc = -1, percA;
    dtSub dt, tdt;
    unsigned long int coun = 0;
    double probEst;

    if ((fe = new fstream("sistema2.txt", ios::in | ios::binary)) == 0) {
        cout << "\nErro ao abrir o arquivo do sistema 2";
        return;
    }
}

```

```

//i
if ((fs = new fstream("tmp2.txt", ios::out)) == 0) {
    cout << "\nErro ao abrir arquivo";
    return;
}
//f

fe->seekg(0, ios::end);
tamT = fe->tellg();
fe->seekg(0);

cout << "\n";
GS2 = 0;
do {
    probEst = 1;

    //le as 12 dimensoes do estado e calcula a probabilidade limite
    for (i = 0; i < 12; i++) {
        *fe >> tmp;
        if (fe->eof()) {
            break;
        }
        dt.v[i] = tmp;

        probEst *= ((powl(cheSub2[i], tmp)) / (powl(saiSub2[i], tmp) *
            tabFat[tmp]));
    }

    //codigo para mostrar a percentagem do arquivo lida
    tamL = fe->tellg();
    percA = perc;
    perc = (int) (((float) tamL / tamT) * 100);
    if (perc == 0) perc = 100;
    if (perc != percA) {
        cout << (char)13 << perc << " %";
    }

    if (fe->eof()) {
        break;
    }

    //calcula a probabilidade de bloqueio do estado lido
    for (i = 0; i < 12; i++) {
        tdt = dt;
        tdt.v[i]++;
        if (!vsSub2(&tdt)) {
            blSub2[i] += probEst;
        }
    }

    //i
    *fs << probEst << "\n";
    //f

    GS2 += probEst;
    coun++;
} while (!fe->eof());

for (i = 0; i < 12; i++) {
    blSub2[i] /= GS2;
}

```

```

}

//i
fs->close();
delete fs;

if (((fs = new fstream("tmp2.txt", ios::in | ios::binary)) == 0) ||
    ((ft = new fstream("probLimSub2.txt", ios::out | ios::trunc)) == 0))
{
    cout << "\nErro ao finalizar calculo de probabilidades limite -";
    cout << " Sistema 2";
    return;
}
float ent;

cout << "\nCriando arquivo de probabilidades limite";

while (*fs >> ent) {
    *ft << ent / GS2 << "\n";
}

*ft << "\nG: " << GS2;

ft->close();
fs->close();

if (fs != 0) {
    delete fs;
}
if (ft != 0) {
    delete ft;
}
//f

cout << "\nLidos " << coun << " registros";
fe->close();
delete fe;
}
/*
//funcoes que calculam as probabilidades de bloqueio do sistema propostas
por Zaim (2001)
void probBloSub1(void) {
    vetBlo1[0] = 1 - ((1 - blSub1[0]) * (1 - blSub1[4]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo1[1] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[7]) *
                    (1 - blSub2[5]));
    vetBlo1[2] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
    vetBlo1[3] = 1 - ((1 - blSub1[5]) * (1 - blSub1[8]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo1[4] = 1 - ((1 - blSub1[5]) * (1 - blSub1[7]) * (1 - blSub2[7]) *
                    (1 - blSub2[5]));
    vetBlo1[5] = 1 - ((1 - blSub1[5]) * (1 - blSub1[7]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
    vetBlo1[6] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo1[7] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[8]) *
                    (1 - blSub2[5]));
    vetBlo1[8] = 1 - ((1 - blSub1[9]) * (1 - blSub1[10]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
}

```

```

void proBloSub2(void) {
    vetBlo2[0] = 1 - ((1 - blSub1[0]) * (1 - blSub1[4]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo2[1] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[7]) *
                    (1 - blSub2[5]));
    vetBlo2[2] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
    vetBlo2[3] = 1 - ((1 - blSub1[5]) * (1 - blSub1[8]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo2[4] = 1 - ((1 - blSub1[5]) * (1 - blSub1[7]) * (1 - blSub2[7]) *
                    (1 - blSub2[5]));
    vetBlo2[5] = 1 - ((1 - blSub1[5]) * (1 - blSub1[7]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
    vetBlo2[6] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo2[7] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[8]) *
                    (1 - blSub2[5]));
    vetBlo2[8] = 1 - ((1 - blSub1[9]) * (1 - blSub1[10]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
}

```

//funcoes que calculam as probabilidades de bloqueio do sistema propostas por Zaim et al. (2003)

```

void proBloSub1(void) {
    vetBlo1[0] = 1 - ((1 - blSub1[0]) * (1 - blSub1[1]) * (1 - blSub1[6]) *
                    (1 - blSub1[11]) * (1 - blSub2[4]) * (1 - blSub2[0]));
    vetBlo1[1] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[10]) *
                    (1 - blSub2[6]) * (1 - blSub2[5]));
    vetBlo1[2] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
    vetBlo1[3] = 1 - ((1 - blSub1[5]) * (1 - blSub1[6]) * (1 - blSub1[11]) *
                    (1 - blSub2[4]) * (1 - blSub2[0]));
    vetBlo1[4] = 1 - ((1 - blSub1[5]) * (1 - blSub1[1]) * (1 - blSub1[3]) *
                    (1 - blSub2[10]) * (1 - blSub2[6]) * (1 - blSub2[5]));
    vetBlo1[5] = 1 - ((1 - blSub1[5]) * (1 - blSub1[1]) * (1 - blSub1[3]) *
                    (1 - blSub2[10]) * (1 - blSub2[9]));
    vetBlo1[6] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo1[7] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[4]) *
                    (1 - blSub2[1]) * (1 - blSub2[5]));
    vetBlo1[8] = 1 - ((1 - blSub1[9]) * (1 - blSub1[1]) * (1 - blSub1[6]) *
                    (1 - blSub1[3]) * (1 - blSub2[10]) * (1 - blSub2[9]));
}

```

```

void proBloSub2(void) {
    vetBlo2[0] = 1 - ((1 - blSub1[0]) * (1 - blSub1[1]) * (1 - blSub1[6]) *
                    (1 - blSub1[11]) * (1 - blSub2[4]) * (1 - blSub2[0]));
    vetBlo2[1] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[10]) *
                    (1 - blSub2[6]) * (1 - blSub2[5]));
    vetBlo2[2] = 1 - ((1 - blSub1[0]) * (1 - blSub1[3]) * (1 - blSub2[10]) *
                    (1 - blSub2[9]));
    vetBlo2[3] = 1 - ((1 - blSub1[5]) * (1 - blSub1[6]) * (1 - blSub1[11]) *
                    (1 - blSub2[4]) * (1 - blSub2[0]));
    vetBlo2[4] = 1 - ((1 - blSub1[5]) * (1 - blSub1[1]) * (1 - blSub1[3]) *
                    (1 - blSub2[10]) * (1 - blSub2[6]) * (1 - blSub2[5]));
    vetBlo2[5] = 1 - ((1 - blSub1[5]) * (1 - blSub1[1]) * (1 - blSub1[3]) *
                    (1 - blSub2[10]) * (1 - blSub2[9]));
    vetBlo2[6] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[4]) *
                    (1 - blSub2[0]));
    vetBlo2[7] = 1 - ((1 - blSub1[9]) * (1 - blSub1[11]) * (1 - blSub2[4]) *
                    (1 - blSub2[1]) * (1 - blSub2[5]));
}

```

```

        vetBlo2[8] = 1 - ((1 - blSub1[9]) * (1 - blSub1[1]) * (1 - blSub1[6]) *
                        (1 - blSub1[3]) * (1 - blSub2[10]) * (1 - blSub2[9]));
    }

    */
    //funcoes que calculam as probabilidades de bloqueio do sistema propostas
    por este trabalho
    void proBloSub1(void) {
        vetBlo1[0] = 1 - ((1 - blSub1[4]) * (1 - blSub2[4]));
        vetBlo1[1] = 1 - ((1 - blSub1[3]) * (1 - blSub2[7]));
        vetBlo1[2] = 1 - ((1 - blSub1[3]) * (1 - blSub2[10]));
        vetBlo1[3] = 1 - ((1 - blSub1[8]) * (1 - blSub2[4]));
        vetBlo1[4] = 1 - ((1 - blSub1[7]) * (1 - blSub2[7]));
        vetBlo1[5] = 1 - ((1 - blSub1[7]) * (1 - blSub2[10]));
        vetBlo1[6] = 1 - ((1 - blSub1[11]) * (1 - blSub2[4]));
        vetBlo1[7] = 1 - ((1 - blSub1[11]) * (1 - blSub2[8]));
        vetBlo1[8] = 1 - ((1 - blSub1[10]) * (1 - blSub2[10]));
    }

    void proBloSub2(void) {
        vetBlo2[0] = 1 - ((1 - blSub1[4]) * (1 - blSub2[4]));
        vetBlo2[1] = 1 - ((1 - blSub1[3]) * (1 - blSub2[7]));
        vetBlo2[2] = 1 - ((1 - blSub1[3]) * (1 - blSub2[10]));
        vetBlo2[3] = 1 - ((1 - blSub1[8]) * (1 - blSub2[4]));
        vetBlo2[4] = 1 - ((1 - blSub1[7]) * (1 - blSub2[7]));
        vetBlo2[5] = 1 - ((1 - blSub1[7]) * (1 - blSub2[10]));
        vetBlo2[6] = 1 - ((1 - blSub1[11]) * (1 - blSub2[4]));
        vetBlo2[7] = 1 - ((1 - blSub1[11]) * (1 - blSub2[8]));
        vetBlo2[8] = 1 - ((1 - blSub1[10]) * (1 - blSub2[10]));
    }

    //le arquivo com os valores de cudl e cisl
    void leConf(void) {
        fstream* f = new fstream("conf.txt", ios::in | ios::binary);
        *f >> cudl >> cisl;
        criaTabFat(cudl > cisl ? cudl : cisl);
    }

    //funcao principal do calculo da probabilidades de bloqueio
    //laco principal
    void proBloqueio(void) {
        int i;
        bool sai = true;
        unsigned long int coun = 0;
        //le as taxas de chegada e saida do arquivo
        leTaxas();
        //inicializa as taxas de chergada e saida de cada sistema
        //a partir das taxas lidas do arquivo
        cheSub1[0] = chegada[0];
        cheSub1[1] = chegada[1];
        cheSub1[2] = chegada[2];
        cheSub1[5] = chegada[6];
        cheSub1[6] = chegada[7];
        cheSub1[9] = chegada[11];

        saiSub1[0] = saida[0];
        saiSub1[1] = saida[1];
        saiSub1[2] = saida[2];
        saiSub1[5] = saida[6];
        saiSub1[6] = saida[7];
    }

```

```

saiSub1[9] = saida[11];

cheSub2[0] = chegada[15];
cheSub2[1] = chegada[16];
cheSub2[2] = chegada[17];
cheSub2[5] = chegada[18];
cheSub2[6] = chegada[19];
cheSub2[9] = chegada[20];

saiSub2[0] = saida[15];
saiSub2[1] = saida[16];
saiSub2[2] = saida[17];
saiSub2[5] = saida[18];
saiSub2[6] = saida[19];
saiSub2[9] = saida[20];

//inicializa os dois vetores de probabilidades de bloqueio e
//os vetores das probabilidades de bloqueio dos sistemas com zero
for (i = 0; i < 9; i++) {
    vetBlo1[i] = vetBlo2[i] = blSub1[i] = blSub2[i] = 0;
}

for (i = 9; i < 12; i++) {
    blSub1[i] = blSub2[i] = 0;
}

//laco principal
do {
    //calcula as taxas para os satelites ficticios do sub-sistema 1
    taxaSub1();

    //calcula G e as probabilidades de bloqueio do sub-sistema 1
    calculaGSub1();

    proBloSub1();

    //calcula as taxas para os satelites ficticios do sub-sistema 2
    taxaSub2();

    //calcula G e as probabilidades de bloqueio do sub-sistema 2
    calculaGSub2();

    proBloSub2();

    coun++;

    if ((coun % 100) == 0) {
        cout << "\n" << coun << " iterações";
    }

    //continua enquanto nao convergir
} while (!converge());

cout << "\nPronto";

salva();
}

//salva os dados das probabilidades de bloqueio

```

```

void salva(void) {
    fstream* f;
    int i;
    if ((f = new fstream("prob.txt", ios::out | ios::trunc)) == 0) {
        cout << "\nN\u00e3o foi poss\u00edvel abrir o arquivo de saida";
        return;
    }
    *f << "Probabilidades de bloqueio sub-sistema 1\n\n";
    for (i = 0; i < 12; i++) {
        *f << blSub1[i] << " ";
    }
    *f << "\n\nProbabilidades de bloqueio sub-sistema 2\n\n";
    for (i = 0; i < 12; i++) {
        *f << blSub2[i] << " ";
    }
    *f << "\n\nProbabilidades de bloqueio calculadas\n\n";
    for (i = 0; i < 9; i++) {
        *f << vetBlo1[i] << " ";
    }
    *f << "\n\n";
    for (i = 0; i < 9; i++) {
        *f << vetBlo2[i] << " ";
    }
    *f << "\n\nProbabilidades de bloqueio do sistema: \n\n";
    *f << blSub1[0] << "\n" << blSub1[1] << "\n" << blSub1[2] << "\n";
    *f << vetBlo1[0] << "\n" << vetBlo1[1] << "\n" << vetBlo1[2] << "\n";
    *f << blSub1[5] << "\n" << blSub1[6] << "\n";
    *f << vetBlo1[3] << "\n" << vetBlo1[4] << "\n" << vetBlo1[5] << "\n";
    *f << blSub1[9] << "\n";
    *f << vetBlo1[6] << "\n" << vetBlo1[7] << "\n" << vetBlo1[8] << "\n";
    *f << blSub2[0] << "\n" << blSub2[1] << "\n" << blSub2[2] << "\n";
    *f << blSub2[5] << "\n" << blSub2[6] << "\n" << blSub2[9] << "\n";
    f->close();
    delete f;
}

//le o arquivo com as taxas de chegada e saida
void leTaxas(void) {
    fstream* f;
    if ((f = new fstream("taxas.txt", ios::in | ios::binary)) == 0) {
        cout << "\nErro ao abrir o arquivo de taxas";
        return;
    }
    bool chf = true;
    int c = 0;
    double v;
    while (*f >> v) {
        if (chf) {
            chf = false;
            chegada[c] = v;
        }
        else {
            chf = true;
            saida[c] = v;
            c++;
        }
    }
    delete f;
}

//cria uma tabela de valores de fatoriais

```

```

void criaTabFat(int n) {
    tabFat = new int[n + 1];
    tabFat[0] = 1;
    int i, f = 1;
    for (i = 1; i <= n; i++) {
        f *= i;
        tabFat[i] = f;
    }
}

//verifica se os valores das probabilidades de bloqueio convergiram
bool converge(void) {
    bool ret = true;
    int i;
    for (i = 0; i < 9; i++) {
        if (fabs(vetBlo1[i] - vetBlo2[i]) > 0.000001) {
            ret = false;
            //break;
        }
        cout << "\n(i: " << i << ") vb1: " << vetBlo1[i] << " vb2: "
            << vetBlo2[i];
    }
    return ret;
}

```

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

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

Manuais Técnicos (MAN)

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

Notas Técnico-Científicas (NTC)

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

Relatórios de Pesquisa (RPQ)

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

Propostas e Relatórios de Projetos (PRP)

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

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

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

Programas de Computador (PDC)

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

Pré-publicações (PRE)

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