Thienne Johnson e Mauro Margalho

# Avaliação de Desempenho de Sistemas Computacionais

genio
GEN | Informação Online

gen | LTC

# Avaliação de Desempenho de Sistemas Computacionais

abdr
ASSOCIAÇÃO BRASILEIRA DE DIREITOS REPROGRÁFICOS
Respeite o direito autoral

O GEN | Grupo Editorial Nacional reúne as editoras Guanabara Koogan, Santos, LTC, Forense, Método e Forense Universitária, que publicam nas áreas científica, técnica e profissional.

Essas empresas, respeitadas no mercado editorial, construíram catálogos inigualáveis, com obras que têm sido decisivas na formação acadêmica e no aperfeiçoamento de várias gerações de profissionais e de estudantes de Administração, Direito, Enfermagem, Engenharia, Fisioterapia, Medicina, Odontologia, Educação Física e muitas outras ciências, tendo se tornado sinônimo de seriedade e respeito.

Nossa missão é prover o melhor conteúdo científico e distribuí-lo de maneira flexível e conveniente, a preços justos, gerando benefícios e servindo a autores, docentes, livreiros, funcionários, colaboradores e acionistas.

Nosso comportamento ético incondicional e nossa responsabilidade social e ambiental são reforçados pela natureza educacional de nossa atividade, sem comprometer o crescimento contínuo e a rentabilidade do grupo.

# Avaliação de Desempenho de Sistemas Computacionais

**Thienne de Melo e Silva Mesquita Johnson**
Pós-doutorado em Engenharia Elétrica e de Computação
pela Universidade Estadual de Campinas
Atualmente é pesquisadora visitante da Universidade do Arizona (EUA)

**Mauro Margalho Coutinho**
Doutorado em Engenharia Elétrica pela Universidade Federal do Pará
na Área de Telecomunicações
Coordenador do Curso de Bacharelado em Ciência da Computação
na Universidade da Amazônia

gen | LTC

Os autores e a editora empenharam-se para citar adequadamente e dar o devido crédito a todos os detentores dos direitos autorais de qualquer material utilizado neste livro, dispondo-se a possíveis acertos caso, inadvertidamente, a identificação de algum deles tenha sido omitida.

Não é responsabilidade da editora nem dos autores a ocorrência de eventuais perdas ou danos a pessoas ou bens que tenham origem no uso desta publicação.

Apesar dos melhores esforços dos autores, do editor e dos revisores, é inevitável que surjam erros no texto. Assim, são bem-vindas as comunicações de usuários sobre correções ou sugestões referentes ao conteúdo ou ao nível pedagógico que auxiliem o aprimoramento de edições futuras. Os comentários dos leitores podem ser encaminhados à LTC – Livros Técnicos e Científicos Editora Ltda.

Direitos exclusivos para a língua portuguesa
Copyright © 2011 by
**LTC – Livros Técnicos e Científicos Editora Ltda.**
**Uma editora integrante do GEN | Grupo Editorial Nacional**

Reservados todos os direitos. É proibida a duplicação ou reprodução deste volume, no todo ou em parte, sob quaisquer formas ou por quaisquer meios (eletrônico, mecânico, gravação, fotocópia, distribuição na internet ou outros), sem permissão expressa da editora.

Travessa do Ouvidor, 11
Rio de Janeiro, RJ – CEP 20040-040
Tels.: 21-3543-0770 / 11-5080-0770
Fax: 21-3543-0896
ltc@grupogen.com.br
www.ltceditora.com.br

Capa: Olga Loureiro

Editoração Eletrônica: Trio Studio

**CIP-BRASIL. CATALOGAÇÃO-NA-FONTE**
**SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ**

J65a

Johnson, Thienne, 1976-
Avaliação de desempenho de sistemas computacionais / Thienne de Melo e Silva Mesquita Johnson, Mauro Margalho Coutinho. - Rio de Janeiro : LTC, 2011.
197p. : il. ; 24 cm

Apêndice
Inclui bibliografia e índice
ISBN 978-85-216-1864-5

1. Desempenho - Avaliação. 2. Padrões de desempenho. 3. Computação - Avaliação. I. Coutinho, Mauro Margalho, 1968-. II. Título.

11-1494. CDD: 004.2
CDU: 004.05

Dedico este livro ao tio Arlindo (*in memoriam*), pelo apoio incondicional, motivação ética, pessoal e profissional.

Thienne Johnson

Dedico este livro à minha amada família, minha esposa Lia e meu filho Thomas. Fontes de minha motivação e alegria de viver.

Mauro Margalho Coutinho

# Prefácio

A área de avaliação de desempenho tem sido um verdadeiro divisor de águas na vida de muitos profissionais e de acadêmicos, tanto na graduação quanto na pós-graduação. Fatores como a escassez de bibliografia em português e o uso de uma abordagem em que se pressupõe uma grande quantidade de conhecimentos já adquiridos acabam por frustrar os interessados que se sentem impotentes diante de tantos pré-requisitos. Este livro foi pensado e concebido tendo como proposta de público-alvo os iniciantes em avaliação de desempenho. A metodologia utilizada se baseia na apresentação de conceitos associados a aplicações práticas, facilitando a assimilação e incentivando o exercício construtivo do saber. Obviamente, o escopo teve que ser reduzido em prol de uma abordagem mais gradativa de aprendizado, na qual o leitor irá consolidar seus conceitos sem se sobrecarregar da teoria inerente à área.

Essa foi uma das motivações para aplicarmos grande parte dos exemplos à área de redes de computadores. Trata-se, portanto, de um texto introdutório que faz uso de muitas associações, exemplos e exercícios que buscam fixar conceitos, preparando o leitor para uma fase que exige mais dedicação. O livro está estruturado em quatro partes. A primeira explora os conceitos básicos, apresentando a taxonomia de avaliação de desempenho; a segunda aborda modelagem, focando principalmente nas aplicações baseadas em teoria de filas; a terceira apresenta as técnicas de simulação, abordando o *Network Simulator*, um dos mais conceituados simuladores de Rede; e a quarta aborda a técnica de aferição, comparação de sistemas e estudos de caso, completando o conjunto básico de conhecimentos introdutórios necessários à inserção do leitor no mundo da avaliação de desempenho.

*Os Autores*

# Agradecimentos

Deixamos expressos nossos sinceros agradecimentos às seguintes instituições e pessoas, sem as quais o presente trabalho teria sido impossível:

- a Deus, por todas as bençãos que eu e minha família recebemos;
- à minha família, por todo o apoio nos meus estudos e motivação para continuar a crescer pessoal e profissionalmente;
- à minha mãe que, além de tudo, ainda me ajudou a revisar o texto deste livro;
- ao amigo Mauro Margalho, que teve a ideia deste projeto do livro e me ajudou a alcançar este objetivo;
- ao Professor Luis Carlos Trevelin, que me apresentou a arte da avaliação de desempenho;
- ao Grupo GEN e a Ricardo Redisch, que acreditaram neste projeto.

Thienne Johnson

- a Deus pela proteção contínua e incondicional;
- à Universidade da Amazônia, minha segunda casa há 15 anos;
- aos amigos Fabrícia de Nazaré Freitas Costa e Lailson Carlos de Andrade Cardoso pelo apoio técnico nas aferições realizadas com o IPERF;
- à minha amiga Thienne Johnson pelo incentivo ao desenvolvimento deste trabalho;
- aos amigos Prof. MSc. Edvar Luz, Prof. DSc. Ádamo Santana e Prof. DSc. Diego Cardoso que sempre me acolheram com afeto e amizade;
- ao meu coqueiro Perseus pelos momentos de reflexão;
- à minha esposa Lia pelo estímulo, amizade, carinho, críticas, sugestões e paciência;
- ao Grupo GEN e a Ricardo Redisch, que acreditaram neste projeto.

Mauro Margalho Coutinho

# Material Suplementar

Este livro conta com materiais suplementares.

O acesso é gratuito, bastando que o leitor se cadastre em http://gen-io.grupogen.com.br.

GEN-IO (GEN | Informação Online) é o repositório de material suplementar e de serviços relacionados com livros publicados pelo GEN | Grupo Editorial Nacional, o maior conglomerado brasileiro de editoras do ramo científico-técnico-profissional, composto por Guanabara Koogan, Santos, LTC, Forense, Método e Forense Universitária.

# Sobre os Autores

**Thienne de Melo e Silva Mesquita Johnson** possui graduação em Tecnologia em Processamento de Dados pela Universidade da Amazônia, mestrado em Ciência da Computação pela Universidade Federal de São Carlos, doutorado em Ciência da Computação pela Universidade Federal de Pernambuco e Pós-doutorado pela Universidade Estadual de Campinas. Foi professora da Universidade da Amazônia, do curso de bacharelado em Ciência da Computação e da Universidade de São Paulo, do curso de Sistemas de Informação. Tem experiência na área de Ciência da Computação, com ênfase em Redes de Computadores e Avaliação de Desempenho, atuando principalmente nos seguintes temas: computação móvel e pervasiva, redes sem fio e redes de sensores. Atualmente é pesquisadora visitante da Universidade do Arizona (EUA).

**Mauro Margalho Coutinho** possui graduação em Tecnologia em Processamento de Dados pela Universidade Federal do Pará, mestrado em Ciência da Computação pela Universidade Federal de Pernambuco na área de redes de computadores e doutorado em Engenharia Elétrica pela Universidade Federal do Pará na área de Telecomunicações. É professor titular, pesquisador, membro efetivo do comitê de Ética e coordenador do curso de bacharelado em Ciência da Computação na Universidade da Amazônia. Compõe o conselho editorial da Revista Traços. Atua principalmente nos seguintes temas: computação móvel, redes tolerantes a desconexões, redes wireless e avaliação de desempenho.

# Sumário

**Parte I** Conceitos Básicos, 1

**Capítulo 1** Introdução à Avaliação de Desempenho, 3

1.1 O que É Avaliação de Desempenho, 3
1.2 Avaliação de Sistemas Computacionais, 4
1.3 Técnicas de Avaliação, 4
*1.3.1 Modelagem, 5*
*1.3.2 Simulação, 6*
*1.3.3 Experimentação ou Aferição, 6*
*1.3.4 Híbrido, 6*
1.4 Definições Básicas, 7
*1.4.1 Métricas, 7*
*1.4.2 Parâmetros, 7*
*1.4.3 Carga de Trabalho, 8*
*1.4.4 Exemplos, 9*
1.5 Quantidades Básicas, 9
*1.5.1 Utilização, 9*
*1.5.2 Vazão, 10*
*1.5.3 Tempo de Resposta, 11*
1.6 Exercícios, 12

**Parte II** Modelagem, 13

**Capítulo 2** Teoria de Filas, 15

2.1 Introdução, 15
2.2 Sistemas de Fluxo, 16
*2.2.1 Classificação dos Sistemas de Fluxo, 16*
2.3 Identificando o Processo Estocástico, 17
2.4 Caracterização dos Sistemas de Fila Única, 18
*2.4.1 Variáveis Relacionadas a uma Fila, 19*
*2.4.2 Medidas e Relacionamento entre as Variáveis, 20*
*2.4.3 Notação de Kendall, 22*
*2.4.4 Exemplos, 24*
2.5 Análise de Fila Única, 25
*2.5.1 Fila M/M/1 – O Sistema Clássico, 25*

*2.5.2 Fila M/M/m – m-Servidores, 28*
*2.5.3 Fila M/M/∞ – Infinitos Servidores, 31*
*2.5.4 Fila M/M/1/B – Capacidade Finita, 33*
*2.5.5 Fila M/M/m/B – m-Servidores com Capacidade Finita, 36*
*2.5.6 Fila M/M/1//K – População Finita com um Servidor, 39*
*2.5.7 Fila M/M/∞//K – População Finita com Infinitos Servidores, 41*
*2.5.8 Fila M/M/m/B/K – m-Servidores com Capacidade e População Finitos, 44*
2.6 Exercícios, 46

**Capítulo 3** Análise Operacional de Fila Isolada, 49

3.1 Introdução, 49
3.2 Tipos de Centro de Serviço, 49
3.3 Variáveis Operacionais, 51
3.4 Leis Fundamentais e Teoremas Operacionais, 52
*3.4.1 Lei da Utilização, 52*
*3.4.2 Lei de Little, 54*
*3.4.3 Lei do Tempo de Resposta Interativo, 55*
*3.4.4 Hipótese do Equilíbrio de Fluxo, 57*
*3.4.5 Teorema da Taxa de Processamento, 58*
*3.4.6 Teorema do Tempo de Resposta, 59*
3.5 Exercícios, 60

**Capítulo 4** Análise Operacional de Redes de Filas, 63

4.1 Introdução, 63
4.2 Classificação, 63
*4.2.1 Tipos de Filas, 64*
*4.2.2 Tipos de Classes de Cargas, 65*
*4.2.3 Número de Classes de Cargas, 65*
4.3 Variáveis Operacionais, 66
4.4 Leis Operacionais para Redes de Filas, 67
*4.4.1 Lei do Fluxo Forçado, 67*
*4.4.2 Lei do Tempo de Resposta, 68*
*4.4.3 Lei da Demanda de Serviço, 70*
4.5 Exercícios, 72

**Capítulo 5** Análise Quantitativa de Redes de Filas, 75

5.1 Introdução, 75
5.2 Análise de Modelos com Carga Única, 75

*5.2.1 Classes Abertas, 75*
*5.2.2 Classes Fechadas, 79*
5.3 Análise de Redes com Múltiplas Cargas, 82
*5.3.1 Classes Abertas, 82*
*5.3.2 Classes Fechadas, 85*
5.4 Exercícios, 85

**Parte III** Simulação, 87

**Capítulo 6** Modelagem via Simulação, 89

6.1 Motivação para Simular, 90
6.2 Conceitos, 91
*6.2.1 Simulações Dependentes de Tempo, 91*

**Capítulo 7** Simulação Aplicada a Redes de Computadores, 93

7.1 Elementos de um Simulador de Redes, 93
7.2 Entendendo o *Network Simulator* (NS-2), 94
*7.2.1 Instalação e Configuração, 95*
*7.2.2 Planejamento da Simulação, 95*

**Capítulo 8** Executando e Analisando a Simulação, 101

8.1 Escrevendo a Simulação, 102
8.2 Executando a Simulação, 106
8.3 Inserindo Adversidades na Simulação, 108
8.4 Inserindo Randomicidade na Simulação, 110

**Parte IV** Aferição, 113

**Capítulo 9** Experimentação ou Aferição, 115

9.1 Motivação para Aferir, 115
9.2 *Benchmarking*, 115
9.3 Utilização do IPERF em Aferição, 117
*9.3.1 Contextualização do Cenário, 118*

**Capítulo 10** Inserindo Credibilidade Estatística aos Resultados de uma Avaliação de Desempenho, 123

10.1 Introdução, 123
10.2 Intervalo de Confiança da Média, 123

*10.2.1 Conceitos Básicos, 124*
*10.2.2 Teorema do Limite Central, 125*
*10.2.3 IC para Amostras Pequenas, 129*
10.3 Comparação de Sistemas, 130
*10.3.1 Teste da Média Zero, 131*
*10.3.2 Observações Pareadas, 132*
*10.3.3 Observações Não Pareadas, 133*
*10.3.4 Teste Visual, 136*
10.4 Exercícios, 138

**Capítulo 11** Estudo de Caso: Modelagem de Sistemas Web, 143

11.1 Introdução, 143
11.2 Transação Web, 144
11.3 Desempenho no Lado Servidor, 144
11.4 Desempenho da Transmissão dos Arquivos, 146
11.5 Desempenho Percebido pelo Cliente, 148
11.6 Exercícios, 148

**Capítulo 12** Avaliação de Desempenho com Virtualização, 151

12.1 Exercícios, 156

**Capítulo 13** Simulação com Redes de Sensores, 157

13.1 Topologias de Redes sem Fio, 157
13.2 Contextualização do Cenário, 158
13.3 Definição da Área de Localização dos Sensores, 162
13.4 Codificando a Simulação, 163
13.5 Exercício, 172

Apêndice, 173
Bibliografia, 177
Índice, 179

# Parte I

# Conceitos Básicos

1 Introdução à Avaliação de Desempenho 3

# Introdução à Avaliação de Desempenho

# 1

## 1.1 O que É Avaliação de Desempenho

O dicionário Houaiss define desempenho sendo a "maneira como atua ou se comporta alguém ou algo, avaliada em termos de eficiência, de rendimento".

Este conceito pode ser utilizado em qualquer aspecto da vida diária, como na avaliação na hora de comprar um novo automóvel, no momento de avaliar o trabalho de um funcionário ou o rendimento dos alunos em uma sala de aula. O Quadro 1.1 mostra alguns exemplos de itens que são medidos no dia a dia e usados para avaliar o dewsempenho e realizar comparações.

**QUADRO 1.1** Exemplos de avaliações

| Item | Medição | O que comparar |
|---|---|---|
| Automóvel | Velocidade máxima; aceleração; consumo: combustível gasto/km rodado. | Qual carro é mais rápido. Qual carro gasta menos combustível. |
| Caixa de banco | Clientes atendidos por minuto. | Qual funcionário é mais produtivo. |
| Restaurante | Mesas ou clientes servidos por hora. | Qual garçom / cozinheiro é mais produtivo. |
| Pedágio na rodovia | Carros por minuto. | Frequência na rodovia em diversos horários. |
| Companhia aérea | Pouso ou decolagem por hora; *check-in* realizados por hora. | Qual horário é mais ocupado. |

A avaliação de desempenho inclui tudo o que é possível mensurar quantitativamente, mas também permite avaliações qualitativas. Por exemplo, pode-se avaliar a satisfação dos usuários quanto à utilização de um software ou transmissão de dados em uma rede de computadores.

O objetivo deste livro é mostrar as técnicas de avaliação de desempenho quantitativas para uso em sistemas computacionais. Nas próximas seções serão vistos os conceitos básicos de avaliação de desempenho de sistemas computacionais.

## 1.2 Avaliação de Sistemas Computacionais

Os sistemas computacionais atendem os usuários de diversas formas. Um sistema de banco de dados recebe requisições dos usuários para realizar algum serviço, como consulta a uma tabela, inserção de dados em um registro, modificações nas tabelas etc. Um sistema de redes de computadores recebe pacotes na interface de rede e deve transmiti-los para outra interface de rede ligada no mesmo meio.

De forma geral, o sistema computacional recebe requisições (pedidos, *jobs*, pacotes etc.) dos clientes (usuários, entidades, softwares etc.) para realizar alguma ação, isto é, realizar algum tipo de serviço para o usuário. A avaliação de desempenho dos sistemas computacionais deverá considerar o tipo de requisições recebidas, o tipo de clientes e o tipo do serviço a ser realizado. O Quadro 1.2 mostra exemplos de sistemas computacionais e os possíveis itens que podem ser medidos e comparados.

**QUADRO 1.2** Exemplos de avaliações em sistemas computacionais

| Tipo de sistema | Medição | O que comparar |
|---|---|---|
| Software cliente | Tempo de inicialização, requisições atendidas por minuto. | Uso do software em diferentes computadores. |
| Redes de computadores | Utilização de banda, vazão de pacotes. | Link de rede mais rápido. |
| Bancos de dados | Consultas atendidas por minuto, tempo de atendimento de consulta. | Qual melhor servidor de banco de dados. |
| Sistemas de arquivos e memória | Taxa de transferência de blocos/páginas. | Melhor sistema operacional para operações físicas. |

## 1.3 Técnicas de Avaliação

Basicamente existem três técnicas de avaliação de desempenho: modelagem, simulação e medição. Cada técnica possui vantagens e desvantagens e devem ser escolhidas segundo alguns critérios[1] que serão vistos a seguir, mas, na prática, o avaliador usa a técnica com a qual sente-se mais familiarizado.

[1] Para um estudo completo sobre o tópico, ver Jain (1991).

A escolha da técnica de avaliação depende muito do status de implementação/construção do sistema a ser avaliado. Para o design de um novo sistema é possível usar simulação ou modelagem. O uso de modelos teóricos ou simulador nos permite a avaliação de diferentes cenários e a comparação entre eles. Imagine o projeto de um sistema de e-commerce. Em um modelo, é possível criar diversas situações de uso, de carga de trabalho, de tempo de processamento de transações etc., e verificar se o servidor consegue atender à demanda de serviço ou se é insuficiente. Portanto, o avaliador terá diversas opções a modificar e então selecionar a configuração mais vantajosa.

No caso da medição, primeiro é necessário já ter o sistema a ser medido ou algum protótipo. Um protótipo pode ser usado nas fases de avaliação para a verificação da melhor configuração, antes que o produto esteja em sua fase final de implementação (ou construção). O custo de realizar diversas modificações em um protótipo será menor do que modificar um produto finalizado. Também é possível usar a medição para comparar um sistema existente com futuras versões deste, quando será possível confirmar se as novas funcionalidades ou modificações levam a um ganho de desempenho e a quantificar esse ganho.

Os modelos de medição também podem ser usados como entrada de um modelo de simulação ou de modelagem, gerando um modelo híbrido (que será visto na Seção 1.3.4).

Para uma maior confiança nos dados avaliados é necessário utilizar duas ou mais técnicas para a verificação e validação dos resultados obtidos (o Capítulo 10 apresentará formas de comparar a avaliação de diferentes sistemas). Têm-se então as seguintes opções:

**Modelagem × Simulação**
**Modelagem × Medição**
**Simulação × Medição**
**Modelagem × Simulação × Medição**

Nas próximas subseções serão vistos conceitos de cada técnica, e ao longo do livro, elas serão abordadas uma a uma, em detalhes.

### *1.3.1 Modelagem*

O uso de modelos torna possível a abstração de detalhes de baixo nível de um sistema, que não são importantes para a caracterização do desempenho de alto nível. Isso faz com que o processo de modelagem seja mais simples do que outras técnicas de avaliação, já que nos permite identificar e eliminar detalhes irrelevantes.

Por exemplo, imagine a avaliação de um sistema de redes de computadores. Os detalhes importantes para saber o desempenho do link são a largura de banda do link, a taxa de geração de pacotes no lado emissor, o tamanho de cada pacote e a taxa de processamento de recepção no lado receptor. Os detalhes, como tipo de camada física e tamanho da memória dos computadores, não precisam ser incluídos em um modelo de alto nível.

No processo de modelagem, portanto, deve-se identificar os principais componentes do sistema, sua forma de interação e algumas suposições de funcionamento para realizar o estudo de avaliação.

Algumas das principais técnicas de modelagem de sistemas computacionais são teoria de filas (Capítulo 2), análise operacional de filas (Capítulos 3 e 4), análise quantitativa de filas (Capítulo 5), processos estocásticos, redes de petri, entre outros.

### 1.3.2 Simulação

A técnica da simulação é uma das mais usadas em sistemas computacionais devido à facilidade de aprendizado e de uso. A simulação pode ser usada para prever o desempenho de um sistema ainda não existente e também verificar como se comporta um sistema existente em diferentes cenários. Para um cenário onde existe a necessidade de modificações em hardware (p. ex., uma rede de computadores ou dos barramentos de uma CPU), a simulação é uma técnica barata para realizar os estudos necessários de desempenho antes/depois de uma modificação.

Entretanto, um problema comum na simulação é a geração de modelos falhos, onde os resultados oferecidos não são corretos. Isto ocorre devido a problemas na especificação do software ou dos modelos estatísticos (p. ex., de geração de carga). Por isso, é recomendável usar simuladores existentes que tenham sido avaliados pela comunidade científica ou profissional e certificados de seu funcionamento correto.

A técnica de simulação será estudada nos Capítulo 6, 7 e 8.

### 1.3.3 Experimentação ou Aferição

Quando um sistema computacional existente é avaliado através de experimentação ou aferição, é possível capturar o desempenho do sistema em dado momento, usando a carga de trabalho em uso atual. Dependendo do tipo de sistema computacional, diversos parâmetros podem ser medidos, como tempo de responder a uma requisição e utilização de recursos.

Um sistema de software pode ter em seu código fonte métodos capazes de capturar informações de desempenho para análise posterior ou em tempo real. Uma rede de computadores pode ter softwares instalados em diversas máquinas que capturam a informação de tráfego e utilização dos links.

Outra forma de estudos de experimentação ou aferição é através do uso de benchmarking. Os benchmarks são softwares usados para comparar o desempenho de sistemas específicos através de um subconjunto de funções daquela aplicação. Por exemplo, os softwares de benchmark para CPU usam um conjunto de carga de trabalho para utilizar todos os recursos do sistema e avaliar o desempenho, geralmente comparando com algum sistema básico. Benchmarking geralmente é usado para avaliar processadores, sistemas de entrada e saída, subsistema de memória, redes de computadores e bancos de dados.

A técnica de experimentação ou aferição será vista no Capítulo 9.

### 1.3.4 Híbrido

Nem sempre é possível realizar o estudo de desempenho de um sistema usando apenas a técnica de medição. Para sistemas ainda incompletos (necessitando a implementação de certas partes), o uso de um modelo híbrido, aliando mais de uma técnica na geração do resultado final, pode ser bastante útil nos casos em que é necessário estudar o comportamento de um sistema existente, mas que necessita ainda da construção de outras partes do sistema.

Também pode-se usar um modelo híbrido quando já se tem resultados de outros tipos de avaliação (p. ex., modelagem) e quer-se analisar o comportamento de outro sistema de mais alto nível, utilizando como entrada os valores gerados pelo modelo de menor nível (p. ex., um

modelo de simulação de uma CPU que possui como parâmetros de entrada os valores de desempenho de um modelo de medição de um barramento).

A Figura 1.1 mostra possíveis combinações entre as técnicas para a geração de um modelo híbrido.

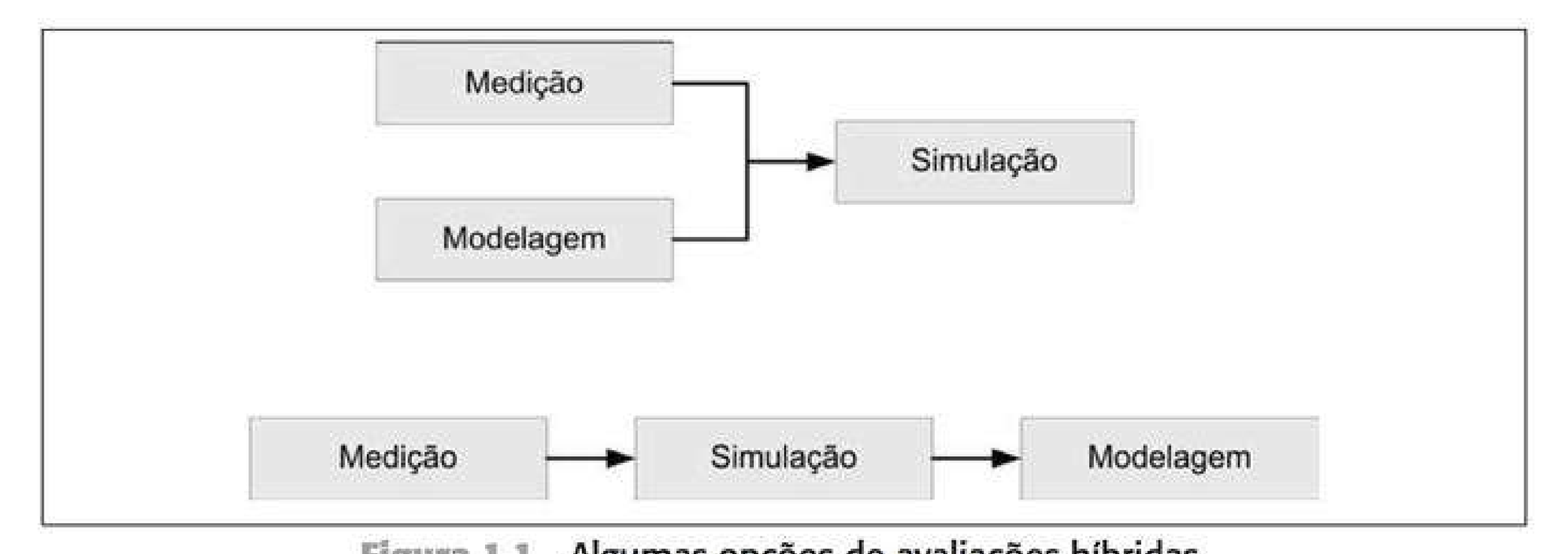


**Figura 1.1** Algumas opções de avaliações híbridas.

## 1.4 Definições Básicas

Algumas definições são necessárias para iniciar o estudo de avaliação de desempenho. Será visto o que são métricas, parâmetros e carga de trabalho, e qual a relação entre eles em um estudo de avaliação de desempenho.

### *1.4.1 Métricas*

Métricas são os critérios usados para quantificar o desempenho de um sistema. Em um estudo de desempenho, geralmente uma ou mais métricas são escolhidas para serem coletadas e comparadas para diferentes cenários.

Suponha um sistema cliente-servidor Web, onde tem-se um servidor atendendo um sistema de e-commerce. Na avaliação, é possível usar a métrica de quanto tempo o sistema gasta para responder a um pedido de compra e comparar quanto tempo essa requisição demora, dependendo do número de itens de cada pedido. Ou pode-se usar essa métrica para comparar o uso de duas máquinas servidoras diferentes. Para uma rede de computadores é desejável verificar a métrica de quanto tempo os pacotes demoram a trafegar na rede, ou o número de pacotes entregues por minuto.

Para cada estudo de desempenho é necessário escolher primeiramente a métrica – o que é necessário avaliar ou comparar. Cada métrica a ser usada depende do tipo de sistema a ser avaliado.

### *1.4.2 Parâmetros*

Após a seleção das métricas, o próximo passo é a escolha dos parâmetros que serão variados durante o estudo de desempenho.

No caso do sistema de e-commerce mencionado anteriormente, os parâmetros eram o número de itens de cada pedido e o uso de duas máquinas servidoras diferentes (isto é, uma configuração de hardware diferente).

No caso de uma rede de computadores e avaliação de um link de comunicação, poder-se-ia ter como parâmetros para comparação o tamanho de um pacote, o protocolo de camada de transporte ou a largura de banda da rede.

Os parâmetros escolhidos em cada estudo devem possuir relação com a métrica escolhida, isto é, afetar de alguma forma a métrica.

Por exemplo, em um sistema de e-commerce onde quer-se avaliar a velocidade de resposta de uma requisição, não importa a quantidade de memória da máquina do cliente, já que a requisição é atendida totalmente na máquina servidora. No caso da rede de computadores, o software que está sendo executado na máquina emissora não irá afetar a velocidade de transmissão no link de comunicação.

No decorrer do livro, será visto, em cada tipo de sistema, que parâmetros escolher, dependendo da métrica avaliada.

### *1.4.3 Carga de Trabalho*

A carga de trabalho (ou *workload*) é a lista de requisições ao sistema, isto é, a quantidade de requisições que o sistema tem para atender.

Em uma rede de computadores, a carga de trabalho é o número de pacotes que a rede deve transmitir. Em um sistema de banco de dados, a carga é o número de requisições de consulta. Em um computador, a carga pode ser o número de instruções que a CPU tem que processar ou o número de acessos à memória solicitados.

Nos estudos de avaliação de desempenho, a escolha da carga de trabalho correta é crucial para garantir que os resultados sejam aceitáveis. A escolha de uma carga de trabalho inapropriada ou não representativa pode levar o estudo de desempenho a ser considerado inaceitável ou incorreto.

A carga de trabalho deve ser representativa de cenários correspondentes aos cenários de uso reais, tanto em termos de quantitativos quanto qualitativos.

A carga de trabalho quantitativamente correta reflete o uso do sistema computacional a ser usado em um cenário real. A escolha de uma carga considerada ideal, mas abaixo das expectativas de uso real, levará o estudo de desempenho a resultados enganosos e subdimensionados.

Suponha o sistema de e-commerce, onde o estudo de atendimento de requisições prevê que ele atenderá 100 clientes por hora, porém uma carga real seria de 400 requisições por hora. O resultado do estudo diria que o sistema está atendendo com grande velocidade de resposta, mas no uso real, o servidor estaria sobrecarregado e atendendo com velocidade de resposta insatisfatória.

Da mesma forma, a escolha de uma carga muito acima das expectativas levará a um resultado superdimensionado. Porém, dependendo do objetivo do estudo de desempenho, esta segunda opção poderá ser a escolhida.

No caso do servidor de e-commerce, o estudo que usa como carga 1000 clientes por hora mostrará que o servidor terá velocidade satisfatória. Este é um caso de estudo superdimensionado, mas o avaliador tinha como objetivo exatamente verificar se o superdimensionamento da carga atenderia futuras cargas de trabalho mais altas do que as atuais.

A carga de trabalho qualitativamente incorreta também afeta o estudo de desempenho no momento em que se escolhe tipos diferentes de carga do que as realmente encontradas. No caso de um servidor de jogos on-line, se o avaliador escolher como carga de trabalho usuários com perfil de usuários de bibliotecas on-line ele terá um dimensionamento e resultados completamente diferentes do correto para sua avaliação.

### *1.4.4 Exemplos*

#### CPU

- Objetivo: aumentar o número de requisições atendidas pelo processador.
- Métrica: requisições atendidas por segundo.
- Parâmetros: tamanho da requisição, tipo da instrução de máquina.
- Carga: requisições do sistema operacional em uso contínuo da máquina por um usuário comum.

#### Banco de dados

- Objetivo: verificar que tipo de consulta é atendida mais rapidamente.
- Métrica: tempo de atendimento de requisição.
- Parâmetro: tipo da requisição SQL (consulta, inserção e deleção de registros).
- Carga: requisições de usuários do sistema bancário durante horário de atendimento externo.

#### Rede de computadores

- Objetivo: verificar o tipo de rede mais adequado para uma aplicação de folha de pagamento dentro de uma empresa.
- Métricas: utilização da banda e pacotes transmitidos por segundo.
- Parâmetros: uso de redes diferentes (Wi-Fi e Ethernet) com diferentes largura de banda e MTU (unidade máxima de transmissão).
- Carga: requisições de funcionários do RH e do Financeiro.

## 1.5 Quantidades Básicas

Nesta seção serão vistas algumas das métricas e parâmetros mais usados nos estudos de avaliação de desempenho. Eles serão usados ao longo do livro nos mais diversos exemplos e estudos de caso.

### *1.5.1 Utilização*

A utilização é a fração do tempo em que o recurso permanece ocupado atendendo às requisições feitas pelos usuários. Essa métrica é representada em porcentagem (%).

**Exemplo 1**

Um servidor Web fica ligado 24 horas por dia, mas atende clientes somente das 6h às 24h, então o servidor está ocupado apenas durante 18 horas por dia. Portanto,

$$U = \frac{18}{25} = 0{,}75 = 75\%$$

**Exemplo 2**

Um sistema de banco de dados fica disponível 20 horas por dia, mas só é usado no horário comercial, das 8h às 18h. Portanto,

$$U = \frac{10}{20} = 0{,}5 = 50\%$$

### *1.5.2 Vazão*

Vazão é a taxa na qual os pedidos recebidos são atendidos pelo sistema computacional. Por taxa entende-se o número de pedidos atendidos por unidade de tempo.

A vazão (ou *throughput*) é uma das métricas mais usadas em sistemas computacionais para avaliar quantitativamente os sistemas existentes no dia a dia. Exemplos comuns de uso da métrica vazão em sistemas computacionais são:

- CPUs: MIPS ou MFLOPS.
- Redes de computadores: bits por segundo (bps).
- Sistemas de bancos dados: consultas solicitadas por segundo.

**Exemplo 1**

Um link de transmissão entre duas máquinas foi monitorado para verificar o desempenho de duas máquinas diferentes ligadas a esse link. A máquina A gerou no link 3 milhões de pacotes em um segundo. A máquina B gerou 7 milhões de pacotes em dois segundos. Qual máquina teve a maior vazão para o link?

$$\text{Vazão} = \frac{\text{Bits}}{\text{segundo}}$$

$$\text{Vazão}_A = \frac{3.000.000}{1} = 3 \text{ Mbps}$$

$$\text{Vazão}_B = \frac{7.000.000}{2} = 3{,}5 \text{ Mbps}$$

Portanto, a máquina B teve maior vazão no link do que a máquina A.

**Exemplo 2**

Um Java Servlet foi instalado em duas máquinas diferentes. Pretende-se testar qual máquina é mais rápida para processar os pedidos recebidos dos usuários. A máquina A atendeu a cinco requisições por segundo, e a máquina B atendeu a duas requisições por segundo. Portanto, a máquina A tem maior vazão do que a máquina B em requisições atendidas por segundo.

### *1.5.3 Tempo de Resposta*

Tempo de resposta é o tempo decorrido entre o pedido da realização de serviço pelo cliente e a chegada da resposta do pedido realizado até o cliente requisitante.

Esta métrica é usada geralmente como a medida de velocidade de um sistema. Por exemplo, ao comparar dois links de comunicação, o mais rápido é o que teve menor tempo para transmitir um pacote. Para comparar dois computadores, o menor tempo de resposta para realizar um processamento nos informa qual a máquina mais rápida.

A Figura 1.2 mostra o conceito de tempo de resposta graficamente.

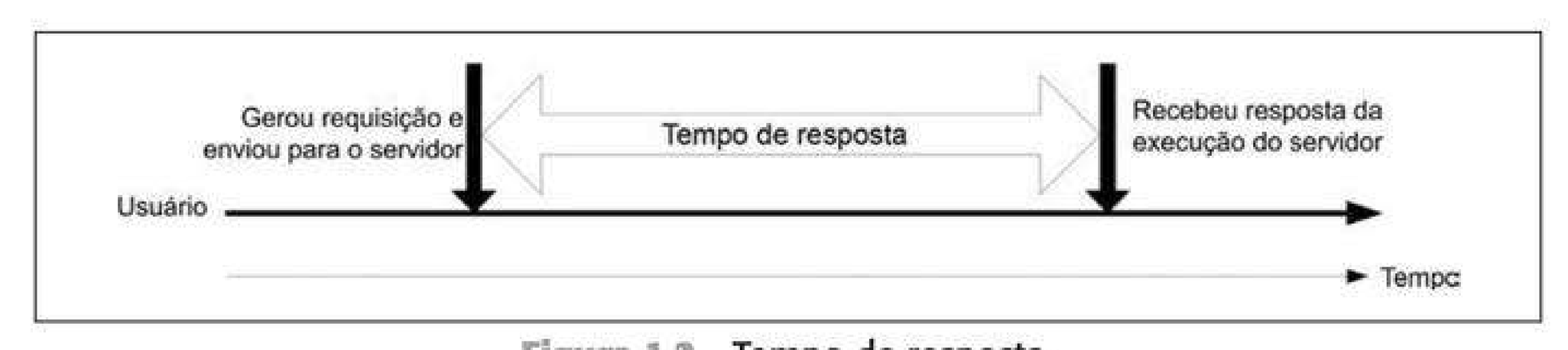


**Figura 1.2** Tempo de resposta.

Existem autores que consideram o tempo de resposta também o tempo entre a realização de um pedido e o início da geração da resposta, mas neste livro será adotada apenas a chegada da resposta do pedido no cliente.

**Exemplo 1**

Uma requisição de impressão foi solicitada às 20h1min00s e a impressão terminou às 20h02min30s. O tempo de resposta é de 1min30s.

**Exemplo 2**

Uma consulta a um banco de dados é realizada pelo cliente às 23h59min00s. A resposta foi recebida às 00h00min10s do dia seguinte. O tempo de resposta é de 70 segundos.

## 1.6 Exercícios

1) Caracterize as métricas de desempenho vazão e utilização para os seguintes sistemas:

   a) Automóvel.
   b) Elevador.
   c) Drive de CD-ROM.
   d) Servidor Web.
   e) Hub.

2) Caracterize a carga de trabalho para os seguintes sistemas:

   a) Telefonia celular de transmissão de voz.
   b) Telefonia celular de transmissão de dados.
   c) Sistema de controle de estoque.
   d) Sistema de controle de tráfego aéreo.

3) A métrica tempo de resposta possui um momento inicial e um momento final para a contagem do tempo. Para os seguintes sistemas, identifique em que momento deve-se começar e terminar o tempo de resposta.

   a) Um usuário envia uma consulta a um banco de dados.
   b) Um host A envia um pacote IP em uma subrede para um host B.
   c) Uma memória cache recebe um conjunto de instruções para serem processadas.

4) O computador A possui tempo de resposta para processar um conjunto de instruções igual a 0,03 segundo. O computador B possui tempo de resposta para processamento de instruções equivalentes igual a 0,005 segundo. Qual computador possui melhor desempenho?

5) Um sistema computacional foi observado durante sete dias e verificou-se que, em média, o sistema estava sendo utilizado por 18 horas a cada dia. Qual a utilização do sistema durante os sete dias?

6) Um link de uma subrede para a Internet possui largura de banda de 4 Mbps (megabits por segundo). Deseja-se atualizar o link para se obter um desempenho 75% maior em relação à vazão. Qual o novo valor de vazão desejado?

# Parte II
# Modelagem


2 Teoria de Filas 15

3 Análise Operacional de Fila Isolada 49

4 Análise Operacional de Redes de Filas 63

5 Análise Quantitativa de Redes de Filas 75

# Teoria de Filas 2

## 2.1 Introdução

Filas fazem parte do dia a dia de todos nós. Enfrentamos filas no supermercado, parque de diversões, na farmácia etc. A Figura 2.1 exemplifica um tipo de fila muito comum, a fila no banco. Clientes entram no banco e se dirigem ao caixa. Quando o funcionário do caixa está atendendo um cliente, os clientes que chegam devem entrar na fila e aguardar sua vez. Quando um cliente termina seu atendimento, ele se dirige à saída e outro cliente, que estava no primeiro lugar da fila, se dirige ao caixa.

Em diversos lugares enfrentamos o mesmo tipo de fila, como no *check-in* do aeroporto e fila de carros no lava a jato.

Nos sistemas computacionais também usamos os sistemas de filas no dia a dia, como, por exemplo, na hora de mandar arquivos para impressão quando esta é compartilhada entre diversos usuários. Nesse caso, os pedidos de impressão devem ser recebidos pela impressora ou servidor de impressão, que irá processar um pedido de cada vez, e pegar o próximo pedido da fila para a próxima impressão. Na rede de computadores os pacotes chegarão a um roteador com múltiplas interfaces de rede, e para cada interface, o dispositivo possui uma fila de pacotes para serem processados e reencaminhados.

**Figura 2.1** Fila em um banco.

Neste capítulo estudaremos os sistemas de filas e aprenderemos como estimar diversos aspectos relacionados, como o tempo de atendimento de uma requisição, tempo de espera na fila, número de clientes dentro de um sistema, entre outros.

## 2.2 Sistemas de Fluxo

Sistemas de filas representam um tipo de classe de eventos dinâmicos, chamados de sistemas de fluxo. Um sistema de fluxo é um sistema no qual alguma mercadoria[1] flui, se move ou é transferida através de um ou mais canais de capacidade limitada, de forma a ir de um ponto a outro no sistema.

Como exemplo, imagine uma rede de computadores, onde o computador A transfere dados para o computador B, ou uma CPU transferindo bits entre memória e disco. Nesses exemplos as mercadorias seriam os pacotes e os bits; os canais seriam o link de rede e o barramento da CPU.

Os sistemas de fluxo de capacidade limitada podem transferir certa quantidade de mercadoria em um determinado intervalo de tempo, a uma taxa finita. Nos exemplos mostrados, os canais de transferência têm uma capacidade limitada de quantas mercadorias estão sendo transferidas em dado momento, sendo limitados, nesses casos, pelo hardware da máquina.

### *2.2.1 Classificação dos Sistemas de Fluxo*

Os sistemas de fluxo são classificados em fluxos estáveis e não estáveis (Figura 2.2).

Os sistemas estáveis são aqueles nos quais o fluxo passa de forma previsível, isto é, a quantidade de fluxo que passa pelo canal é constante e conhecida durante o intervalo de tempo de observação do sistema. Também são constantes o tempo que o fluxo chega ao canal e quanto aquele fluxo demanda do canal.

Os sistemas estáveis podem ser classificados em canal único ou redes de canais.

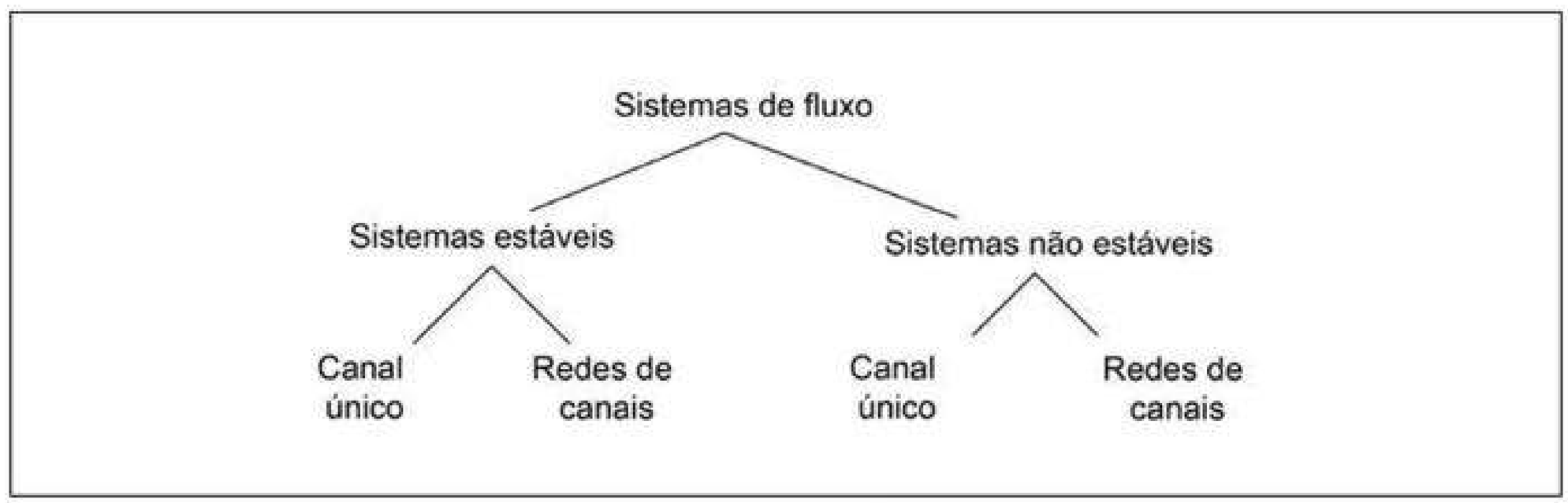


**Figura 2.2** Divisão dos sistemas de fluxo.

[1] Na literatura também encontramos o termo *commodity*.

Nos sistemas de canal único, as mercadorias entram no sistema de um único canal para serem processadas (transferidas, executadas, servidas) pelo canal de capacidade (ou taxa de serviço) **C**. Considerando uma taxa de chegadas (número de mercadorias que chegam durante certo intervalo de tempo) ***R***, temos que:

- Se $C > R$, o sistema consegue processar as mercadorias sem que ocorra sobrecarga do canal.
- Se $R > C$, significa que o canal não consegue processar as mercadorias e a sobrecarga levará o canal a um nível de congestionamento, isto é, filas de mercadorias se formarão.

Os sistemas estáveis de canal único atendem $C > R$. Nos sistemas de redes de canais em sistemas estáveis, temos que $C > R$ em cada canal.

Os sistemas não estáveis são problemas de fluxos estocásticos ou randômicos, isto é, as chegadas das mercadorias ao canal e o tamanho da demanda por uso do canal são imprevisíveis. Como exemplo, temos os sistemas computacionais onde usuários mandam requisições em tempos imprevisíveis, podendo encontrar o serviço solicitado já ocupado por requisições de outros usuários, ou livre para atender àquela requisição imediatamente. Portanto, essa incerteza se estende ao tempo em que a requisição demorará a ser atendida.

Sistemas não estáveis também podem ser divididos em canal único e redes de canais. Os sistemas de fluxo randômicos não são facilmente modelados como os sistemas de fluxo determinístico. A teoria de filas se propõe a ser uma ferramenta para resolver problemas relacionados aos sistemas de fluxo randômicos. É o que veremos a partir da próxima seção.

## 2.3 Identificando o Processo Estocástico

Para a análise dos sistemas de fluxo, incluindo as filas, é necessário identificar o processo estocástico que descreve o fluxo, a estrutura e a disciplina do canal. Considerando um sistema de filas, chamaremos as mercadorias de clientes, e o canal de centro de serviço.

O processo de chegada de clientes no centro de serviço é descrito em termos da distribuição de probabilidade dos tempos entre chegadas dos clientes. Denominado por $A(t)$, temos:

$$A(t) = P\,[\text{tempo entre chegadas} \leq t]$$

significando que a taxa de chegadas de clientes no sistema é determinada pela probabilidade de que o tempo entre chegadas seja menor do que um tempo $t$.

A segunda quantidade é a demanda das chegadas no canal, também chamada de tempo de serviço, isto é, a quantidade de tempo que um cliente demora sendo atendido dentro de um centro de serviço. A distribuição de probabilidade é dada por $B(x)$, tal que:

$$B(x) = P\,[\text{tempo de serviço} \leq x]$$

significando que o tempo de serviço para atendimento de uma requisição do cliente no sistema é determinado pela probabilidade de que o tempo de serviço seja menor do que um tempo $x$. O tempo entre chegadas e o tempo de serviço são variáveis independentes entre si.

Para caracterizar completamente um sistema de filas é necessário definir outras quantidades diversas, como, por exemplo, o tamanho permitido de clientes na fila, número de centros de serviço etc. Veremos na próxima seção a caracterização completa de um sistema de fila única.

## 2.4 Caracterização dos Sistemas de Fila Única

O sistema da Figura 2.3 é composto de uma linha de entrada, o sistema de fila e uma linha de saída.

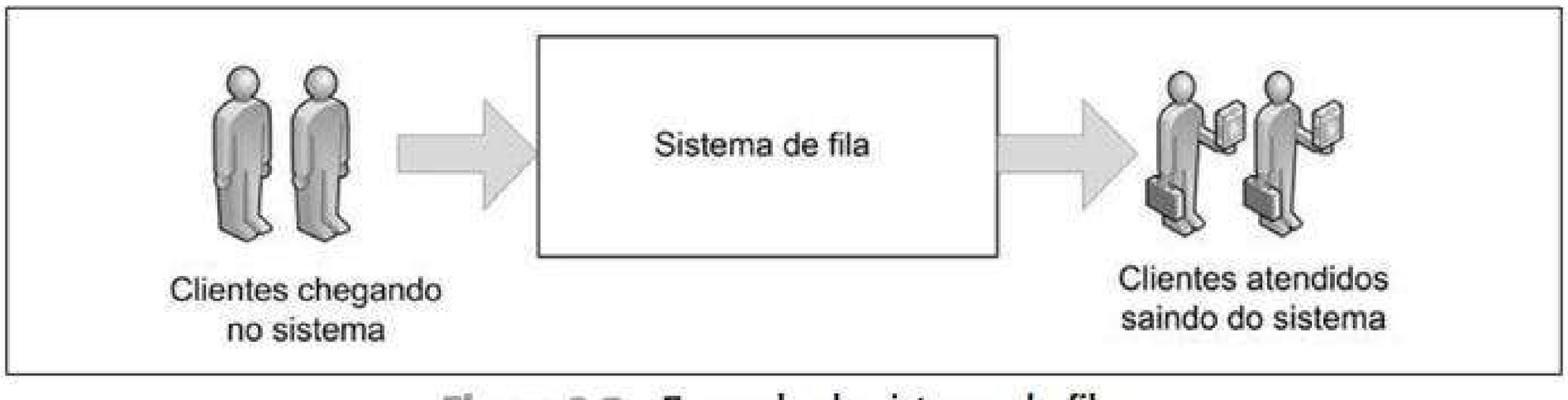


**Figura 2.3** Exemplo de sistema de fila.

Na linha de entrada chegam os clientes (requisições, pacotes, *jobs* etc.) com o tempo arbitrário entre chegadas $A(t)$, que se encaminham para o sistema de filas, onde são atendidos (servidos) em um tempo arbitrário de serviço $B(x)$ e depois são liberados em uma linha de saída.

O sistema de fila encaminha os clientes para um ou mais centros de serviço. Dizemos que o sistema tem $m$ servidores sendo $m \geq 1$. A Figura 2.4 mostra exemplos de uma fila única com $m = 1$ (1 fila e 1 servidor) e $m = 3$ (1 fila e 3 servidores).

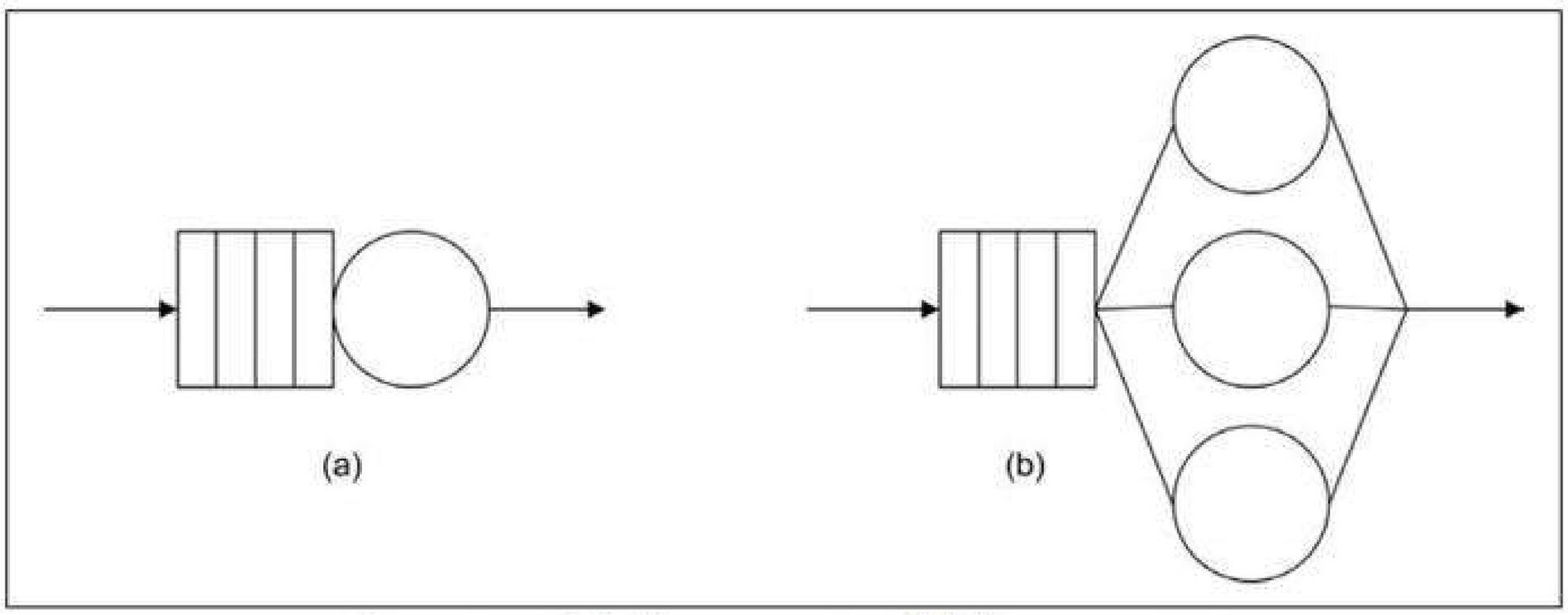


**Figura 2.4** (a) Fila com $m = 1$; (b) fila com $m = 3$.

## 2.4.1 Variáveis Relacionadas a uma Fila

Algumas variáveis são importantes e devem ser analisadas para cada sistema de fila.

### Número de clientes

- $C_n$ denota o $n$-ésimo cliente que entra no sistema.
- $n_w$ representa o número de usuários esperando para serem atendidos. Este número é sempre menor do que $n$, dado que não inclui os usuários que estão sendo atendidos.
- $n_s$ representa o número de usuários em atendimento dentro do centro de serviço.
- $n$ denota o número de clientes no sistema em um determinado instante de tempo. Esta variável inclui os usuários que já estão sendo atendidos, assim como os que estão esperando na fila. A média do número total de clientes no sistema inclui a média dos clientes na fila e a média dos clientes em atendimento.

$$E[n] = E[n_w] + E[n_s]$$

### Taxa de chegadas

A taxa de chegadas ($\lambda_n$) é o tempo de chegada de $C_n$ clientes no sistema. Também representa o número médio de chegadas de clientes por unidade de tempo,[2] sendo o inverso do tempo entre chegadas:

$$\lambda = \frac{1}{A(t)}$$

Por exemplo, se observamos oito chegadas de clientes durante um tempo de observação de quatro minutos, então a taxa de chegadas é $8/4 = 2$ clientes por minuto.

### Taxa de serviço

A taxa de serviço ($\mu$) é a taxa média de serviço por cliente no servidor, isto é, quantos clientes o sistema consegue atender em uma unidade de tempo. É o inverso do tempo de serviço:

$$\mu = \frac{1}{B(x)}$$

Para $m$ servidores conectados a uma mesma fila, a taxa total de serviço é $m\mu$.

Por exemplo, se o servidor do sistema demora dois segundos para atender a uma requisição, então a taxa de serviço é $1/5 = 0{,}5$ requisição por segundo. Se a fila receber mais um servidor para atender os clientes, a taxa de serviço será então uma requisição por segundo.

### Tempo de espera

O tempo de espera ($w$) é o tempo que o cliente espera na fila antes de ser atendido pelo servidor.

[2] Em alguns sistemas dependentes de carga (número de usuários que já se encontram no sistema), esta taxa pode ser uma função do estado do sistema.

### Tempo de resposta

O tempo de resposta (***s***) é o tempo médio total gasto por um cliente, contando desde a entrada do cliente no sistema, incluindo o tempo médio gasto na fila e o tempo médio que demorou em atendimento (***x***), até a sua saída.

$$E[s] = E[w] + E[x]$$

Por exemplo, o servidor demora em média um segundo para atender a uma requisição. O tempo médio na fila para cada requisição é de cinco segundos. Portanto, o tempo médio de resposta para uma requisição seria de seis segundos.

A Figura 2.5 mostra a relação entre algumas das variáveis e uma fila única.

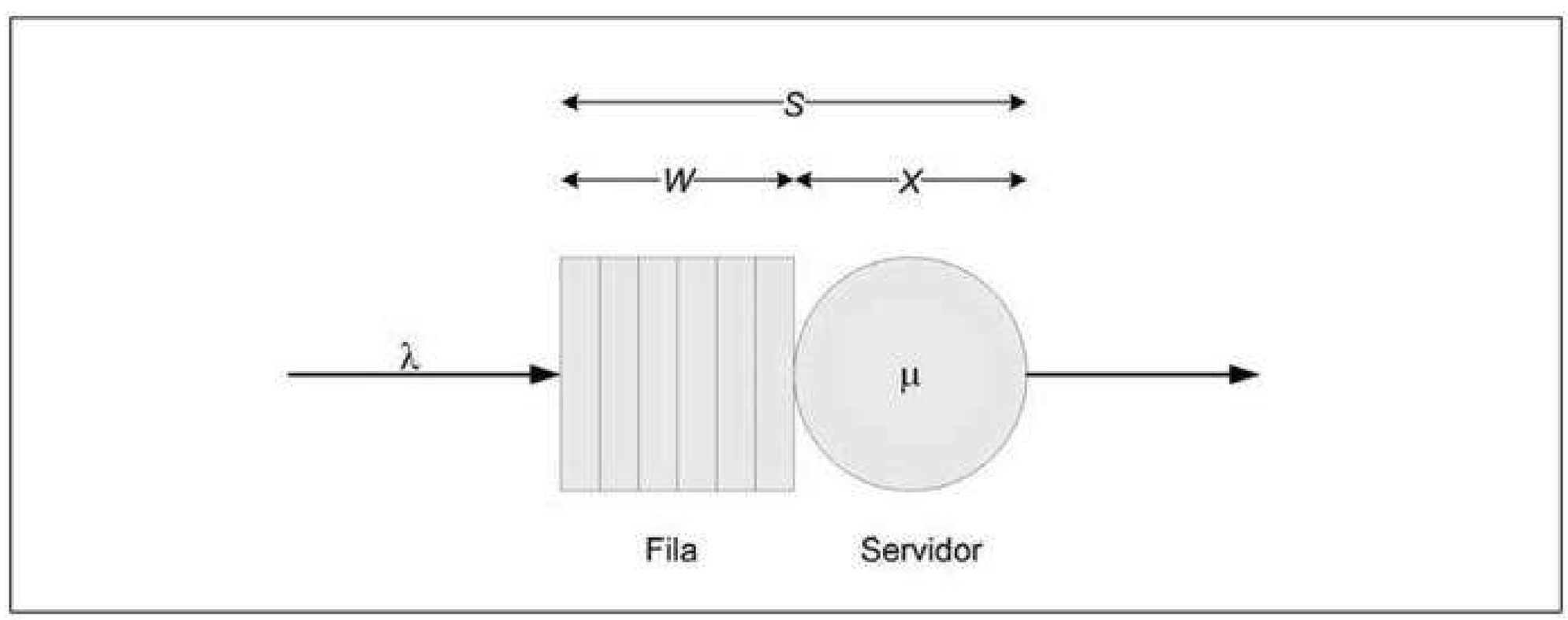


**Figura 2.5** Tempos e variáveis de uma fila única.

## *2.4.2 Medidas e Relacionamento entre as Variáveis*

### Lei de Little

Se os usuários não forem perdidos por causa de *buffers* insuficientes, temos que o número médio de usuários no sistema é a taxa de chegadas multiplicada pelo tempo médio de resposta:

$$E[n] = \lambda * E[s]$$

Da mesma forma, o número médio de usuários na fila é obtido pela taxa de chegadas multiplicada pelo tempo médio de espera:

$$E[n_w] = \lambda * E[w]$$

## Probabilidade do Número de clientes no sistema

Representada por $P_n$, esta probabilidade é utilizada para descrever o comportamento de um sistema de filas em termos do número de clientes, $n$, atualmente no sistema. Para cada tipo de fila, é necessário calcular $P_n$, que pode ser usado para calcular outras informações sobre a fila.

## Utilização

$\boldsymbol{U(t)}$ denota a utilização do sistema, isto é, a fração de tempo na qual o servidor está ocupado. Quando $\boldsymbol{U(t) = 0}$, o sistema está vazio. Quando $\boldsymbol{U(t) > 0}$, o sistema está com alguma ocupação, e em $\boldsymbol{U(t) = 1}$, o sistema está 100% do tempo ocupado.

O servidor é dito ocioso quando não há nenhuma requisição no sistema; em todos os demais momentos ele é dito ocupado. O intervalo de tempo entre dois intervalos ociosos sucessivos é denominado período ocupado.

## Condição de estabilidade

A taxa média de chegadas deve ser menor do que a taxa média de atendimento, ou seja, $\lambda < m\mu$, com $m \geq 1$. Esta relação é válida apenas para filas infinitas. Com filas finitas o sistema nunca é instável.

## Intensidade de tráfego

A intensidade de tráfego ($\rho$) mede o congestionamento do sistema de fila. Se for perto de zero, há pouca fila, e em geral, quando a intensidade de tráfego aumenta, a fila também aumenta. Veremos que, para alguns sistemas de filas, a intensidade de tráfego é equivalente à utilização do sistema.

Definimos a intensidade de tráfego como:

$$\rho = \frac{\text{tempo_médio_serviço}}{\text{tempo_médio_entre_chegadas}} = \frac{\text{taxa_chegadas}}{\text{taxa_serviço}} = \frac{\lambda}{\mu}$$

Esta variável também pode ser usada para formular a condição de estabilidade, que deve ser $\rho < 1$, isto é, a média dos números de requisições que chegam em uma unidade de tempo deve ser menor do que o número de requisições que podem ser processadas. Para $m$ servidores conectados à mesma fila, de forma geral,

$$\rho = \frac{\lambda}{m\mu}.$$

Veremos, para cada tipo de fila, seu valor de $\rho$.

### Vazão

Em termos de filas, vazão é a média das requisições processadas em uma unidade de tempo, isto é, a taxa de saídas do sistema de filas. Para filas em equilíbrio, a taxa de saídas é igual à taxa de chegadas, então podemos definir a vazão como:

$$\lambda = m\rho\mu$$

Para casos onde a fila tem capacidade finita, a vazão poderá ser diferente da taxa de chegadas no sistema.

## 2.4.3 Notação de Kendall

Em 1953, D. G. Kendall criou uma notação composta de três parâmetros funcionais para caracterizar os sistemas de fila. Hoje em dia, a notação possui seis parâmetros e segue o formato A/S/m/B/K/SD, onde:

- **A** é a distribuição do processo de chegadas,
- **S** é a distribuição do tempo de serviço,
- **m** é o número de servidores,
- **B** é a capacidade do sistema,
- **K** é o tamanho da população e
- **SD** é a disciplina de atendimento da fila.

Conheceremos agora cada um desses parâmetros.

### Distribuição do processo de chegadas (A)

É a distribuição do processo de chegadas de clientes ao sistema. Algumas das distribuições mais usadas são (Quadro 2.1):

**QUADRO 2.1** Principais distribuições

| | |
|---|---|
| $M$ | Processo de chegadas de Poisson (ou randômico). |
| $E_k$ | Distribuição de Erlang de parâmetro $k$. |
| $D$ | Tempo entre chegadas fixo ou determinístico. |
| $G$ | Distribuição genérica. |
| $GI$ | Distribuição genérica com tempo entre chegadas independente. |

### Distribuição do tempo de serviço (S)

É a distribuição do tempo de serviço para atendimento de um cliente em um servidor. As principais distribuições usadas são as mesmas para o processo de chegadas (Quadro 2.1).

## Número de servidores (m) ou atendentes

Representa o número de servidores (ou canais de serviço) disponíveis no sistema. No mínimo, o sistema de filas deve ter um servidor, e no máximo, infinitos. Centros de serviço com infinitos servidores são chamados de centros de atraso.

## Capacidade do sistema (B)

A capacidade do sistema é o número máximo de clientes que são permitidos dentro do sistema em certo momento. Quando a capacidade chega ao limite máximo, novos clientes que chegam ao sistema são recusados. Na notação de Kendall, se esse número é omitido, assume-se que a capacidade do sistema é ilimitada.

A capacidade inclui o número máximo de clientes que podem ser atendidos e o número de clientes que cabem na fila. Se esse valor é omitido, significa que a capacidade do sistema é infinita.

## Tamanho da população (K)

É o tamanho da população de clientes, representando o número de clientes interessados em usar o sistema ou possíveis usuários. Por exemplo, o servidor de aplicações acadêmicas de uma universidade tem como possível população alunos, funcionários e professores. Já o servidor de uma aplicação Web tem uma população teoricamente infinita.

Se esse valor é omitido na definição da fila, significa que a população é considerada infinita.

## Disciplina de atendimento da fila (SD)

Representa a ordem em que os clientes são atendidos após chegar à fila. Algumas das principais disciplinas de atendimento são (Quadro 2.2):

**QUADRO 2.2** Algumas disciplinas de atendimento

| Símbolo | Nome | Descrição |
|---|---|---|
| FIFO/FCFS | *First In First Out/First Come First Served* (o primeiro que chega é o primeiro a ser atendido). | Os clientes são atendidos (servidos) na ordem em que chegam. |
| LIFO/LCFS | *Last in First Out/Last Come First Served* (o último que chega é o último a ser atendido). | Os clientes são atendidos (servidos) na ordem reversa à ordem em que chegam. |
| SIRO | *Service In Random Order* (serviço em ordem randômica). | Os clientes são atendidos em ordem randômica, sem considerar a ordem em que chegam. |

*(continua)*

| | | |
|---|---|---|
| RR | *Round Robin*. | Cada cliente tem uma fatia de tempo para ser atendido. Após usar sua fatia de tempo, sua requisição para de ser servida e retorna para o fim da fila (FCFS). Isto se repete até a requisição ter o serviço completado. |
| PS | *Processor Sharing* (compartilhamento de processador). | Equivale a RR com fatias de tempo infinitesimais. Aparenta ter todas as requisições atendidas simultaneamente, mas com o tempo de serviço aumentado. |
| IS | *Infinite Server* (infinitos servidores). | São tantos servidores que nunca se forma fila. |

Se o código da disciplina for omitido na definição da fila, assume-se o uso da disciplina de fila FIFO.

### 2.4.4 Exemplos

Vejamos alguns exemplos do uso da notação de Kendall.

**Exemplo 1 Fila M/M/1**

Fila com processo de chegadas de Poisson, tempo de serviço exponencial e um servidor. Valores omitidos: capacidade do sistema e população ilimitados, disciplina de atendimento FIFO.

**Exemplo 2 Fila M/M/3/30/3000/SIRO**

Fila com processo de chegadas de Poisson, tempo de serviço exponencial, três servidores, capacidade do sistema de 30 clientes, população igual a 3000 clientes, disciplina de fila SIRO.

**Exemplo 3 Fila M/M/1//100**

Fila com processo de chegadas de Poisson, tempo de serviço exponencial, um servidor, capacidade do sistema de 30 clientes, população igual a 100 clientes.
Valores omitidos: capacidade do sistema ilimitada, disciplina de fila FIFO.
✓ Notação opcional: M/M/1/∞/100 ou M/M/1/∞/100/FIFO

**Exemplo 4 Fila M/M/1/10**

Fila com processo de chegadas de Poisson, tempo de serviço exponencial, um servidor, capacidade do sistema de 10 clientes. Valores omitidos: população ilimitada, disciplina de fila FIFO.
✓ Notação opcional: M/M/1/10/∞ ou M/M/1/10/∞//FIFO

## 2.5 Análise de Fila Única

O modelo de filas mais simples contém apenas uma fila, como mostrado na Figura 2.5. Esse modelo pode ser usado para analisar recursos individuais em sistemas de computação. Muitas filas podem ser modeladas como processos de nascimento e morte em que o estado do sistema é representado pelo número de usuários no sistema.

As filas markovianas são filas únicas com chegada e atendimento pelo processo de Poisson. Nesta seção conheceremos as filas únicas markovianas mais comuns.

### 2.5.1 *Fila M/M/1 – O Sistema Clássico*

A fila M/M/1 possui processo de chegadas de Poisson, tempo de serviço exponencial e apenas um servidor, com capacidade e população ilimitada. Ela é modelada como um processo de nascimento e morte em que as taxas se mantêm constantes independentemente do número de usuários:

$$\lambda_n = \lambda, \qquad n = 0, 1, 2, \ldots, \infty$$
$$\mu_n = \mu, \qquad n = 1, 2, \ldots, \infty$$

Portanto,

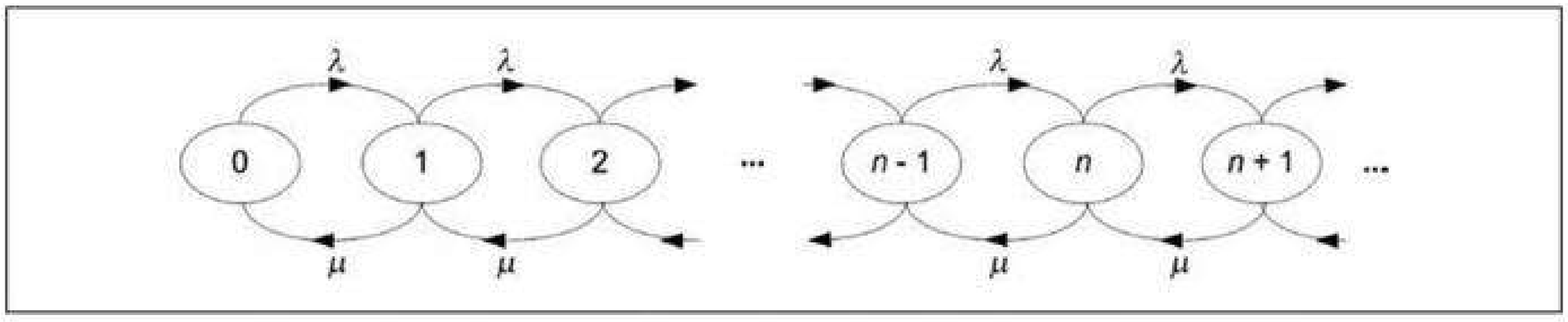


**Figura 2.6** Processo de nascimento e morte para fila M/M/1.

De acordo com a solução de equilíbrio geral,[3] obtemos do processo de nascimento e morte a expressão geral $P_n$ que expressa todas as probabilidades de equilíbrio em termos de uma única constante $P_0$ (a probabilidade de a fila estar vazia, ou 0 cliente no sistema):

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0, \qquad n = 1, 2, \ldots, \infty$$

Sendo $\rho = \frac{\lambda}{\mu}$, e como as condições de estabilidade necessitam de $0 \leq \rho < 1$, garantimos que $p_0 > 0$, então:

$$p_n = \rho^n p_0, \qquad n = 1, 2, \ldots, \infty$$

[3] As deduções referentes aos processos de nascimento e morte podem ser vistas na referência Kleinrock (1976).

Somando todas as probabilidades[4] de $p_n$ para todo $n$ e $\lambda < \mu$, podemos obter $p_0$:

$$p_0 = \frac{1}{1+\sum_{n=1}^{\infty}\left(\frac{\lambda}{\mu}\right)^n} = \frac{1}{1+\sum_{n=1}^{\infty}\rho^n} = \frac{1}{1+\rho+\rho^2+\cdots+\rho^\infty} = 1-\rho$$

Substituindo na expressão para $p_n$, obtemos:

$$p_n = (1-\rho)\rho^n, \qquad n = 0, 1, 2, \ldots, \infty$$

## Outras propriedades da fila M/M/1

- Utilização do servidor (probabilidade de ter uma ou mais requisições no sistema):

$$U = 1 - p_0 = \rho$$

De outra forma, se o sistema for observado durante um tempo $T$,
- O número total de requisições que chegarão e receberão serviço será $\lambda T$.
- O tempo total ocupado do servidor para atender essas requisições será $\lambda T$.

A utilização será então:

$$U = \frac{\text{tempo ocupado}}{\text{tempo total}} = \frac{\lambda T/\mu}{T} = \frac{\lambda}{\mu} = \rho$$

- Número médio de requisições no sistema:

$$E[n] = \sum_{n=1}^{\infty} np_n = \sum_{n=1}^{\infty} n(1-\rho)\rho^n = \frac{\rho}{1-\rho}$$

- Probabilidade de ter $n$ ou mais requisições no sistema:

$$P(\geq n \text{ requisições no sistema}) = \sum_{i=n}^{\infty} p_i = \sum_{i=n}^{\infty} (1-\rho)\rho^i = \rho^n$$

- Tempo médio de resposta para uma requisição:

$$E[s] = \frac{1}{\mu(1-\rho)}$$

[4] Kant (1992).

- Tempo médio de espera na fila de uma requisição:

$$E[w] = E[s] - \frac{1}{\mu} = \frac{1}{\mu(1-\rho)}$$

- Número médio de requisições na fila:

$$E[n_w] = \frac{\rho^2}{1-\rho}$$

**Exemplo**

Um servidor Web conectado a um roteador recebe em média 600 requisições de páginas HTML por minuto. Ele tem capacidade de atender, em média, a 80 requisições por segundo. Calcule todas as propriedades de fila referentes a esse servidor.

Neste exemplo, tem-se um servidor, dotado de uma interface de rede, portando, todas as requisições chegam pela mesma entrada e devem aguardar atendimento em uma fila interna. Tem-se então:

Taxa de chegadas: $\lambda = \frac{600}{60} = 10 \text{ req/s}$

Taxa de serviço: $\mu = 80 \text{ req/s}$

Intensidade de tráfego: $\rho = \frac{\lambda}{\mu} = \frac{60}{80} = 0{,}75$ , portanto, $U = 75\%$

Probabilidade de nenhum usuário no sistema:

$$p_0 = 1 - \rho = 1 - 0{,}75 = 0{,}25 = 25\%$$

Probabilidade de cinco usuários no sistema:

$$p_5 = \rho^n p_0 = 0{,}75^5 \cdot 0{,}25 = 0{,}059 = 5{,}9\%$$

Probabilidade de cinco ou mais usuários no sistema:

$$P(\geq 5 \text{ requisições no sistema}) = \rho^n = 0{,}75^5 = 0{,}24 = 24\%$$

Probabilidade de um ou mais de cinco ou mais usuários no sistema:

$$P(\geq 1 \text{ requisição no sistema}) = \rho^n = 0{,}75^1 = 0{,}75 = 75\%$$

Tempo médio de resposta:

$$E[s] = \frac{1}{\mu(1-\rho)} = \frac{1}{80(1-0{,}75)} = 0{,}05 \text{ s}$$

*(continua)*

Tempo médio de espera na fila:

$$E[w] = \frac{1}{\mu(1-\rho)} = \frac{1}{80(1-0{,}75)} = 0{,}05\text{s}$$

Número médio de usuários:

$$E[n] = \frac{\rho}{1-\rho} = \frac{0{,}75}{1-0{,}75} = 3 \text{ requisições}$$

Número médio de requisições na fila:

$$E[n_w] = \frac{\rho^2}{1-\rho} = \frac{0{,}75^2}{1-0{,}75} = 2{,}25 \text{requisições}$$

## 2.5.2 Fila M/M/m – m-Servidores

Este tipo de fila difere do tipo anterior por possuir um número máximo de *m* servidores idênticos. Neste caso, quando todos os servidores estão ocupados, as requisições são mantidas em uma única fila para todos os servidores (Figura 2.7).

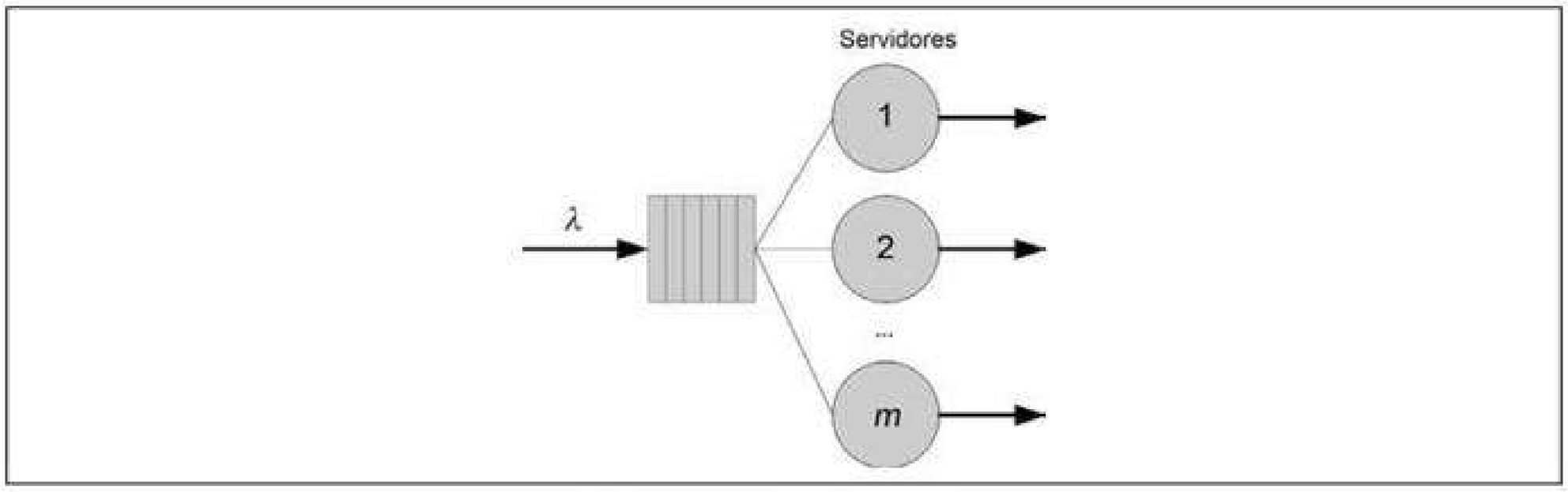


**Figura 2.7** Fila M/M/m.

A taxa de chegadas se mantém constante, mas a taxa de serviço depende do número de requisições que estão no sistema:

$$\lambda_n = \lambda, \qquad n = 0, 1, 2, \ldots, \infty$$

$$\mu_n = \min[n\mu, m\mu] = \begin{cases} n\mu, & 0 \le n < m \\ m\mu, & m \le n \end{cases}$$

Este tipo de fila é modelado de acordo com o seguinte diagrama de transição de estados (Figura 2.8):

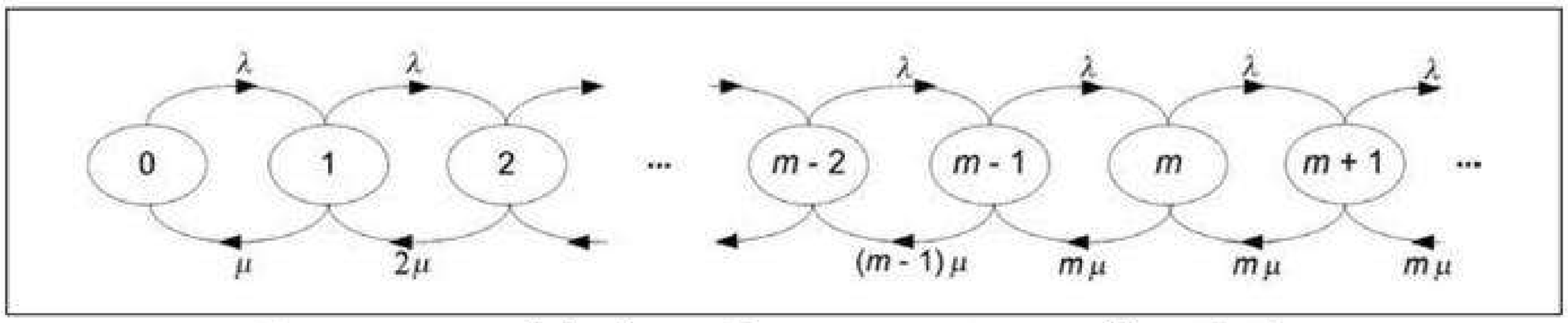


**Figura 2.8** Modelo de nascimento e morte para filas M/M/m.

Neste caso, as equações gerais para probabilidade de $n$ requisições no sistema dependem de os servidores estarem todos ocupados ou não. Portanto, tem-se uma probabilidade para até $n < m$ requisições presentes nos centros de serviço, ou outra probabilidade para quando o número de requisições é maior do que o número de servidores ($n \geq m$):

$$p_n = \begin{cases} \dfrac{1}{n!}\left(\dfrac{\lambda}{\mu}\right)^n p_0, & n \leq m \\ \dfrac{1}{m!\,m^{n-m}}\left(\dfrac{\lambda}{\mu}\right)^n p_0, & n \geq m \end{cases}$$

Reescrevendo $p_n$ em termos de $\rho = \dfrac{\lambda}{m\mu}$, obtém-se:

$$p_n = \begin{cases} \dfrac{(m\rho)^n}{n!} p_0, & n \leq m \\ \dfrac{\rho^n m^m}{m!} p_0, & n \geq m \end{cases}$$

A probabilidade de não existir nenhuma requisição no sistema é calculada por:

$$p_0 = \frac{1}{\dfrac{(m\rho)^m}{m!(1-\rho)} + \displaystyle\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!}}$$

## Outras propriedades da fila M/M/m

- Probabilidade de todos os servidores estarem ocupados:

Quando todos os servidores estiverem ocupados, uma nova requisição que chega ao sistema é forçada a entrar em uma fila. Esta probabilidade é conhecida como fórmula C de Erlang[5] e é dada por:

$$P[fila] = C\left(\mathrm{m}, \rho\right) = p_0 \frac{(m\rho)^m}{m!(1-\rho)}$$

- Número médio de requisições em espera:

$$E[n_w] = \frac{\rho C(m, \rho)}{1-\rho}$$

- Número esperado de requisições em atendimento:

$$E[n_s] = m\rho$$

- Número de requisições no sistema:

$$E[n] = E[n_w] + E[n_s] = m\rho + \frac{\rho C(m, \rho)}{1-\rho}$$

- Utilização:

Se observarmos o sistema por um tempo $T$,

- O número total de requisições que chegarão e receberão serviço será $\lambda T$.
- O tempo total ocupado dos $m$ servidores para atender essas requisições será $\lambda T/m$.

A utilização será então:

$$U = \frac{\text{tempo ocupado por servidor}}{\text{tempo total}} = \frac{(\lambda T/\mu)/m}{T} = \frac{\lambda}{m\mu} = \rho$$

- Tempo médio de resposta:

$$E[s] = \frac{E[n]}{\lambda} = \frac{1}{\mu}\left(1 + \frac{C(m, \rho)}{m(1-\rho)}\right)$$

- Tempo médio de espera na fila:

$$E[w] = \frac{E[n_w]}{\lambda} = \frac{C(m, \rho)}{m\mu(1-\rho)}$$

[5] As deduções usando a fórmula de Erlang são mostradas na referência Jain (1991). Esta relação é representada por alguns autores como $\xi$.

**Exemplo**

O site de uma empresa de RH recebe em média 10 currículos por minuto. Existem duas instâncias de um software de processamento de currículos executando no site, e com isso, a empresa consegue verificar, em média, 12 currículos por minuto em cada instância. Qual o tempo em média que cada currículo fica no sistema, contando o tempo na fila e o tempo de ser processado?

Valores fornecidos no problema:
$m = 2$ servidores (2 instâncias que são os servidores de processamento)
$\lambda = 10$ requisições/minuto
$\mu = 12$ requisições/minuto

$$\rho = \frac{\lambda}{m\mu} = \frac{10}{2\times 12} = 0{,}42$$

Para calcular o tempo de resposta (fila mais atendimento), precisa-se do cálculo de $p_0$ e $C(m, \mu)$, portanto:

$$p_0 = \frac{1}{\dfrac{(m\rho)^m}{m!(1-\rho)} + \sum_{n=0}^{m-1} \dfrac{(m\rho)^n}{n!}} = \frac{1}{\dfrac{(2\times 0{,}42)^2}{2!(1-0{,}42)} + \sum_{n=0}^{2-1} \dfrac{(2\times 0{,}42)^n}{n!}} =$$

$$\frac{1}{\dfrac{0{,}71}{1{,}16} + \dfrac{(2\times 0{,}42)^0}{0!} + \dfrac{(2\times 0{,}42)^1}{1!}} = 0{,}41$$

O próximo passo é calcular $C(m, \mu)$:

$$C(m,\mu) = p_0 \frac{(m\rho)^m}{m!(1-\rho)} = 0{,}41 \frac{(2\times 0{,}42)^2}{2!(1-0{,}42)} = 0{,}21$$

Finalmente,

$$E[s] = \frac{1}{\mu}\left(\frac{C(m,\mu)}{m(1-\rho)}\right) = \frac{1}{12}\left(\frac{0{,}21}{2(1-0{,}42)}\right) = 13{,}92 \text{minutos}$$

O tempo de resposta do sistema é de 13,92 minutos.

### *2.5.3 Fila M/M/∞ – Infinitos Servidores*

Esta fila é um caso especial de filas M/M/m, onde o número de servidores é infinito ($m = \infty$). É possível interpretar o funcionamento desse tipo de sistema como um servidor que sempre aumenta sua taxa de serviço quando um novo cliente chega, ou como um sistema que libera um novo servidor para cada novo cliente. A representação desta fila é mostrada na Figura 2.7, mas com a diferença do servidor $m$ ser $\infty$.

Este tipo de fila também é chamada de **centro de atraso**, em que:

- As requisições nunca têm que esperar.
- O tempo médio de resposta é igual ao tempo médio de serviço, já que o tempo dentro do sistema será apenas o tempo de execução do servidor.

Neste tipo de fila a taxa de chegadas se mantém constante, e a taxa de serviço depende do número de requisições que estão no sistema:

$$\lambda_n = \lambda, \quad n = 0,1,2,\ldots$$
$$\mu_n = n\mu, \; n = 1,2,3,\ldots$$

Este tipo de fila é modelado de acordo com o seguinte diagrama de transição de estados (Figura 2.9):

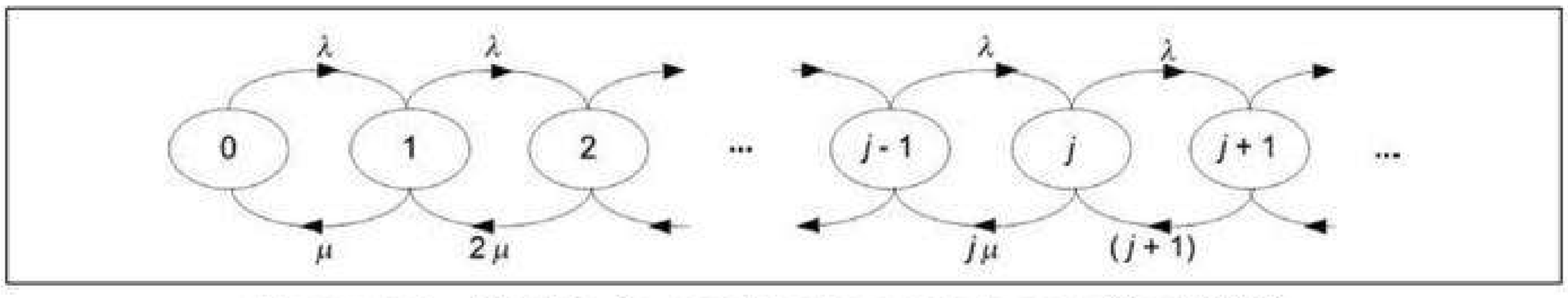


**Figura 2.9** Modelo de nascimento e morte para filas M/M/∞.

Neste tipo de sistema temos:

$$p_n = \frac{\rho^n}{n!} p_0, \qquad n = 1,2,\ldots,\infty$$

onde $\rho = \dfrac{\lambda}{\mu}$

## Outras propriedades da fila M/M/∞

Como existem infinitos servidores, as requisições que chegam ao sistema são prontamente atendidas, portanto, têm-se as seguintes relações diretas ao pronto atendimento.

- Número médio de clientes no sistema:

$$E[n] = \frac{\lambda}{\mu}$$

- Tempo médio de resposta:

$$E[s] = \frac{1}{\mu}$$

**Exemplo**

Um sistema hipotético possui infinitos servidores. Sendo a taxa de chegada de clientes igual a 20 clientes por hora e a taxa de serviço igual a 30 clientes por hora, qual a probabilidade de haver cinco clientes no sistema em um determinado momento dado que a probabilidade de não existirem clientes no sistema é 50%? E qual o número médio de clientes?

Valores fornecidos no problema:
$m$ = infinitos servidores
$\lambda = 20$ clientes/hora
$\mu = 30$ clientes/hora
$p_0 = 50\% = 0{,}50$

$$\rho = \frac{\lambda}{\mu} = \frac{20}{30} = 0{,}67$$

$$p_5 = \frac{\rho^5}{5!}\, p_0 = \frac{0{,}67}{5!} \times 0{,}50 = 0{,}003 = 0{,}3\%$$

$$E\left[n\right] = \frac{\lambda}{\mu} = \frac{20}{30} = 0{,}67 \text{ cliente}$$

O número médio de usuários no sistema é de 0,67 cliente.

## 2.5.4 Fila M/M/1/B – Capacidade Finita

Neste tipo de sistema, a chegada de requisições é limitada por uma capacidade fixa do sistema. O sistema pode suportar a chegada de até $B$ requisições (incluindo a requisição que está em atendimento), e toda nova requisição que chegar enquanto o sistema está com $B$ requisições dentro (na fila e servidor) será descartada, isto é, a entrada será recusada e a requisição partirá sem ter obtido serviço. Este tipo de sistema é chamado de sistema com perdas.

A Figura 2.10 exemplifica onde as requisições são armazenadas dentro do sistema de capacidade limitada.

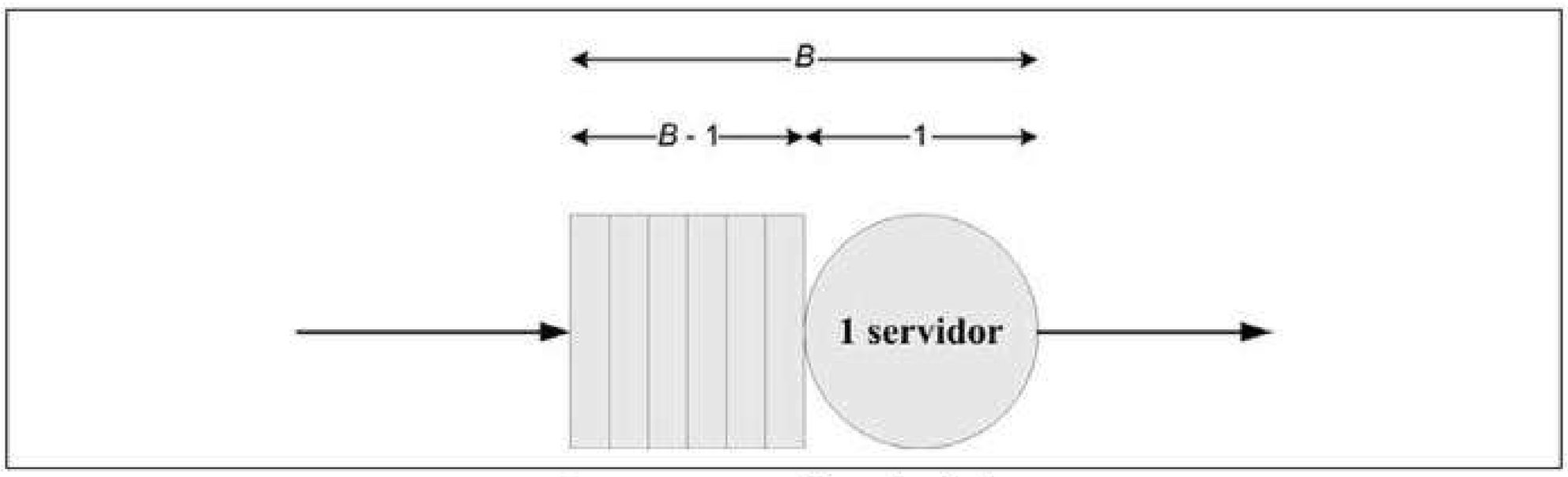


Figura 2.10 Fila M/M/1/B.

Este tipo de fila é modelado de acordo com o diagrama de transição de estados da Figura 2.11.




**Figura 2.11** Modelo de nascimento e morte para filas M/M/1/B.

A taxa de serviço é constante e a taxa de chegadas depende do número de requisições já recebidas e sendo processadas ou na fila:

$$\lambda_n = \begin{cases} \lambda, & n < B \\ 0, & n \geq B \end{cases}$$

$$\mu_n = \mu, \quad n = 1,2,3, \ldots, B$$

As probabilidades de $n$ usuários no sistema se dão por:

$$p_n = \begin{cases} \dfrac{1-\rho}{1-\rho^{B+1}}\,\rho^n, & 0 \leq n \leq B \\ 0, & n > B \end{cases}$$

onde $\rho = \dfrac{\lambda}{\mu}$ e

$$p_0 = \frac{1-\rho}{1-\rho^{B+1}}$$

## Outras propriedades da fila M/M/1/B

- Utilização:

$$U = \rho(1 - p_B)$$

onde $p_B$ é a probabilidade de o sistema estar totalmente cheio. Este tipo de sistema é sempre estável $(\rho < \infty)$.

- Taxa efetiva de chegada:

Como a fila tem capacidade limitada, nem todas as requisições poderão chegar ao sistema. Esta taxa representa os clientes que conseguem entrar na fila antes de a capacidade do siste-

ma não permitir mais entradas. A taxa efetiva então depende da probabilidade de o sistema estar repleto de usuários:

$$\lambda' = \lambda(1 - p_B)$$

- Taxa de perda:

$$\text{Taxa de perda} = \lambda p_B$$

- Número médio de usuários no sistema:

$$E[n] = \frac{\rho}{1-\rho} - \frac{(B+1)\rho^{B+1}}{1-\rho^{B+1}}$$

- Número médio de usuários na fila:

$$E[n_w] = \frac{\rho}{1-\rho} - \rho\,\frac{1+B\rho^{B}}{1-\rho^{B+1}}$$

- Tempo médio de resposta:

$$E[s] = \frac{[En]}{\lambda(1-p_B)}$$

- Tempo médio de espera:

$$E[w] = \frac{[En_w]}{\lambda(1-p_B)}$$

**Exemplo**

Pacotes IP chegam a um roteador de acordo com o processo de Poisson com média de cinco pacotes por segundo. O tempo de encaminhamento de um pacote demora em média 0,1 segundo. O equipamento possui um problema de hardware, e com isso, na fila de entrada só cabem cinco pacotes. Pacotes que chegam na fila lotada são descartados. Calcule a taxa de perdas de pacotes que vão embora sem serviço.

Valores fornecidos no problema:

$m = 1$ servidor
$B = 6$ pacotes (1 sendo servido e 5 na fila)
$\lambda = 5$ pacotes/segundo
$\mu = 1/0{,}1 = 10$ pacotes/segundo

*(continua)*

$$\rho = \frac{\lambda}{\mu} = \frac{5}{10} = 0{,}5$$

A taxa de perdas é calculada por $\lambda p_B$, que é a probabilidade de sistema estar cheio. Portanto,

$$p_B = \frac{1-\rho}{1-\rho^{B+1}}\,\rho^B = \frac{1-0{,}5}{1-0{,}5^7}\,0{,}5^6 = 0{,}937$$

A probabilidade de o sistema estar cheio é de 93,7%. E a taxa efetiva de perdas

$$\text{Taxa de perdas} = \lambda\, p_b = 5 * 0{,}937 = 4{,}685 \text{ pacotes/segundo}$$

## 2.5.5 Fila M/M/m/B – m-Servidores com Capacidade Finita

Este tipo de fila também é um sistema de perdas como a fila M/M/1/B, com a diferença de que, em vez de um servidor, tem-se *m* servidores (Figura 2.12) e uma capacidade limitada a *B* clientes.

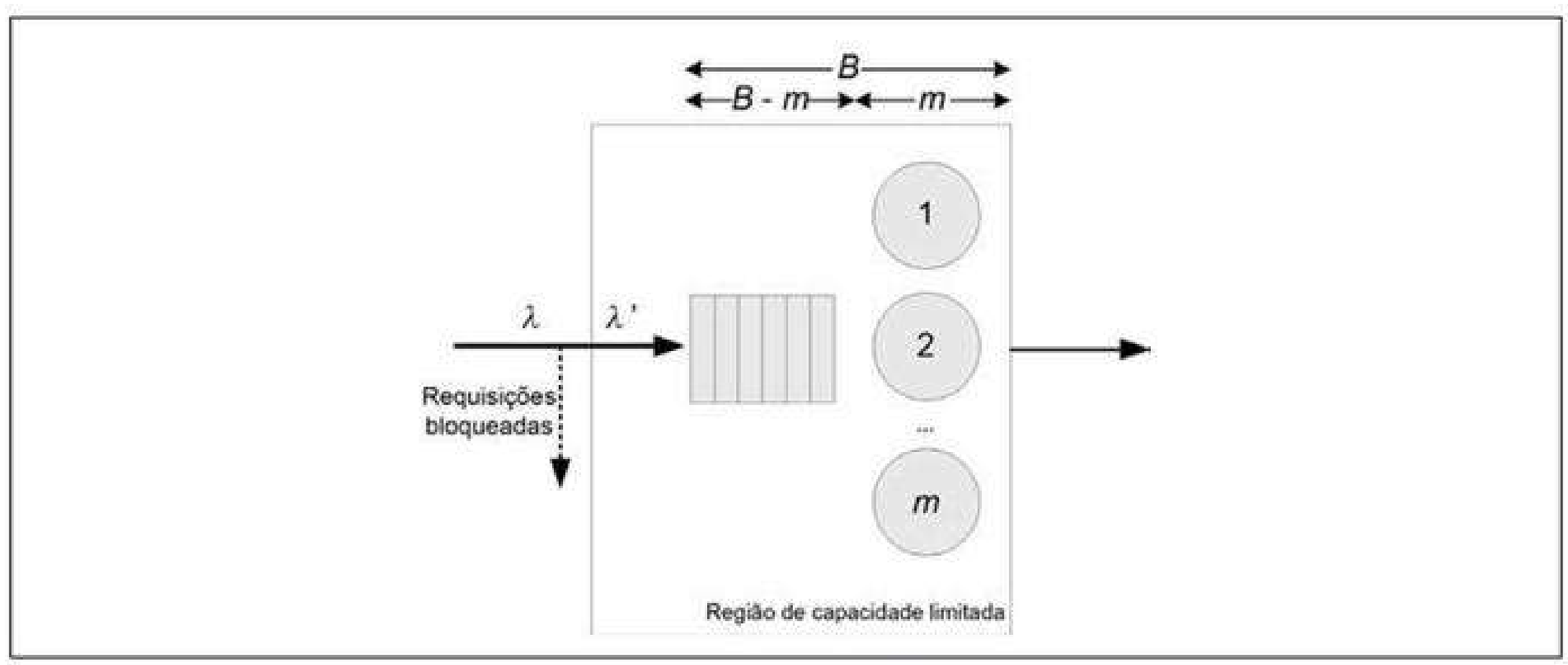


**Figura 2.12** Fila M/M/m/B.

Cada nova requisição que entra no sistema é alocada a um servidor, até que todos os servidores estejam ocupados. Após este limiar, toda nova requisição que chegar entrará na fila, até um limite de *B-m* novos clientes. A taxa de serviço é constante e a taxa de chegadas depende do número de requisições já recebidas e sendo processadas ou na fila, até o limite de *B* requisições:

$$\lambda_n = \begin{cases} \lambda, & n < B \\ 0, & n \geq B \end{cases}$$

$$\mu_n = \begin{cases} n\mu, & n = 1,2,3,\ldots,m-1 \\ m\mu, & n = m, m+1,\ldots,B \end{cases}$$

A Figura 2.13 mostra o diagrama de transição de estados para esta fila.

A probabilidade de $n$ usuários no sistema se dá por:

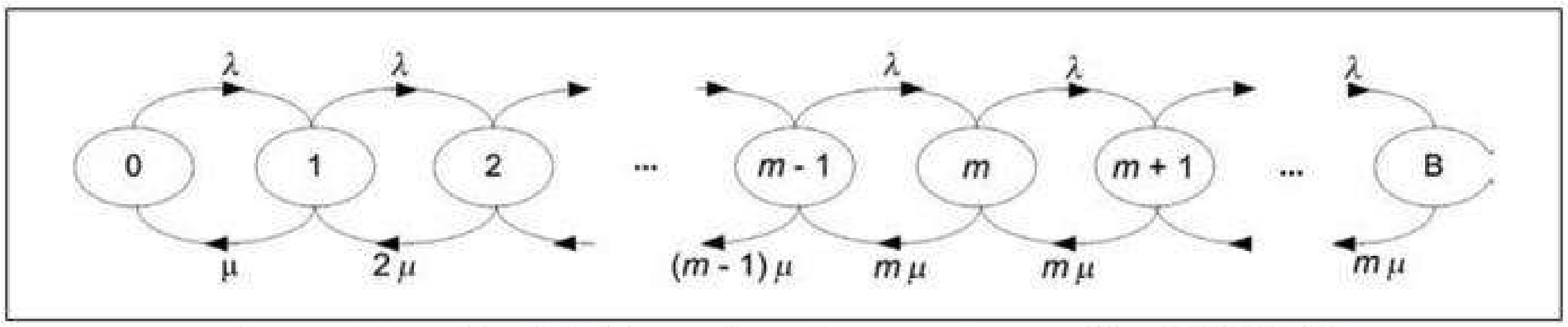


**Figura 2.13** Modelo de nascimento e morte para filas M/M/m/B.

$$p_n = \begin{cases} \left(\dfrac{\lambda}{\mu}\right)^n \dfrac{1}{n!} p_0, & n < m \\ \dfrac{m^m}{m!}\left(\dfrac{\lambda}{\mu}\right)^n p_0, & m \le n \le B \end{cases}$$

e

$$p_0 = \frac{1}{\left[1 + \dfrac{(1-\rho^{B-m+1})(m\rho)^m}{m!(1-\rho)} + \sum_{n=1}^{m-1} \dfrac{(m\rho)^n}{n!}\right]}$$

## Outras propriedades da fila M/M/m/B

- Utilização:

$$U = \rho(1 - p_B)$$

Sendo a intensidade de tráfego $\rho = \lambda / m\mu$.

- Taxa efetiva de chegadas:

$$\lambda' = \lambda(1 - p_B)$$

- Taxa de perda:

$$\text{Taxa de perda} = \lambda p_B$$

- Número médio de usuários no sistema:

$$E[n] = \sum_{n=1}^{B} np_n$$

- Número médio de usuários na fila:

$$E[n_w] = \sum_{n=m+1}^{B} (n-m)p_n$$

- Tempo médio de resposta:

$$E[s] = \frac{[En]}{\lambda(1-p_B)}$$

- Tempo médio de espera:

$$E[w] = \frac{[En_w]}{\lambda(1-p_B)}$$

**Exemplo**

O roteador do exemplo anterior (filas M/M/1/B) recebia em média cinco pacotes IP por segundo e sua taxa de processamento de pacotes era, em média, 10 pacotes por segundo. O equipamento foi atualizado para melhorar o problema de hardware, e com isso, na fila de entrada cabem agora 10 pacotes mas o equipamento consegue processar três pacotes em paralelo. Pacotes que chegam na fila lotada são descartados. Calcule a nova taxa de perdas de pacotes que vão embora sem serviço.

Valores fornecidos no problema:

$m = 3$ servidores
$B = 13$ pacotes (3 sendo servidos e 10 na fila)
$\lambda = 5$ pacotes/segundo
$\mu = 10$ pacotes/segundo

$$\rho = \frac{\lambda}{m\mu} = \frac{5}{3\times 10} = 0{,}17$$

A taxa de perdas é calculada por $\lambda p_B$, que é a probabilidade de o sistema estar cheio. Como $m \leq B \leq B$, tem-se

$$p_0 = \frac{1}{1 + \frac{(1-0{,}17^{13-3+1})}{3!(1-0{,}17)} + \sum_{n=1}^{3-1} \frac{(3\times 0{,}17)^n}{n!}} = 0{,}35$$

e

$$p_B = \frac{m^m}{m!}\left(\frac{\lambda}{\mu}\right)^B p_0 = \frac{3^3}{3!}\left(\frac{5}{10}\right)^{13} 0{,}35 = 0{,}00012$$

A probabilidade de o sistema estar cheio é de 0,012%. E a taxa efetiva de perdas é

$$\text{Taxa de perdas} = \lambda p_b = 5 * 0{,}00012 = 0{,}0006 \text{pacote/segundo}$$

Portanto, o aumento da capacidade do hardware do sistema praticamente anulou a perda de pacotes.

## 2.5.6 Fila M/M/1//K – População Finita com um Servidor

Neste tipo de fila é assumido que a população é finita, isto é, os possíveis usuários do sistema pertencem a um universo finito. A população total é considerada como $K$ clientes. Ao visualizar a Figura 2.14, é possível supor que se tem um cliente no servidor, a fila do servidor tem capacidade ilimitada, mas o número de clientes que podem chegar ao sistema é dado por $K$ - n (população total, excluindo o cliente sendo atendido e os clientes na fila).

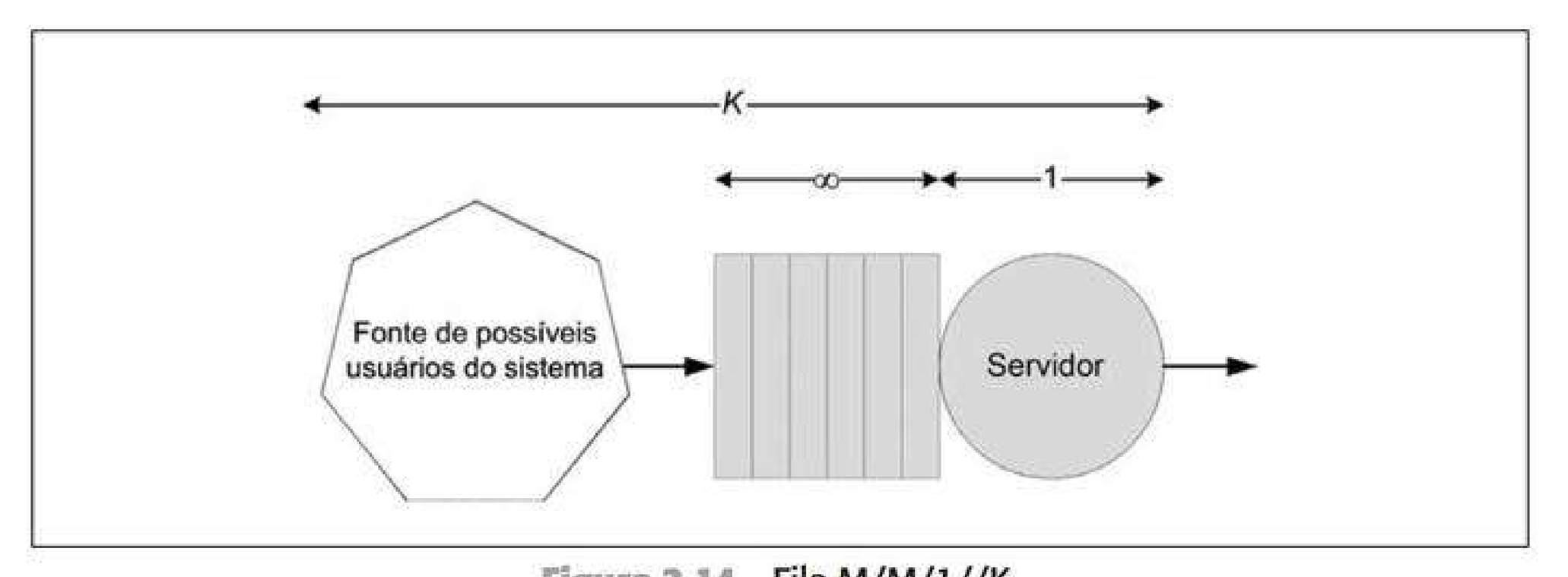


**Figura 2.14** Fila M/M/1//K.

Assim, quando existem $n$ clientes no sistema, existem $K$ - $n$ clientes na fonte da população. Portanto, a taxa de chegadas de clientes é dada por $\lambda(K - n)$. Quanto mais clientes sendo atendidos, menor o número de clientes restantes. Neste caso, clientes chegam ao sistema a uma taxa de chegadas menor, reduzindo futuro congestionamento do sistema.

A taxa de serviço é constante e a taxa de chegadas depende do número de requisições já recebidas e sendo processadas ou na fila, até o limite de $K$ requisições da população:

$$\lambda_n = \begin{cases} \lambda(K-n), & 0 \le n \le K \\ 0, & n > K \end{cases}$$

$$\mu_n = \mu, \; n = 1, 2, \ldots$$

A Figura 2.15 mostra o diagrama de transição de estados para esta fila.

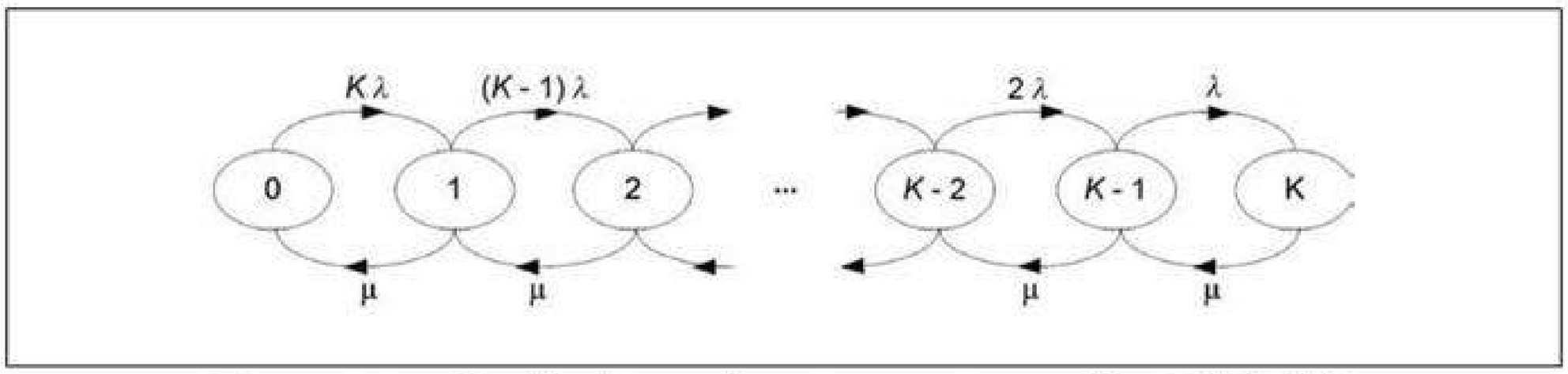


**Figura 2.15** Modelo de nascimento e morte para filas M/M/1//K.

Do diagrama retiramos as probabilidades de $p_n$:

$$p_n = \begin{cases} \left(\dfrac{\lambda}{\mu}\right)^n \dfrac{K!}{(K-n)!} p_0, & 0 \le n \le K \\ 0, & n > K \end{cases}$$

e

$$p_0 = \frac{1}{\sum_{n=0}^{K} \left(\dfrac{\lambda}{\mu}\right)^n \dfrac{K!}{(K-n)!}}$$

## Outras propriedades da fila M/M/1//K

- Utilização:

$$U = \rho(K - E[n])$$

sendo $\rho = \lambda/\mu$

- Número médio de usuários na fila:

$$E[n_w] = K - \left(\frac{\lambda - \mu}{\lambda}\right)(1 - p_0)$$

- Número médio de usuários no sistema:

$$E[n] = E[n_w] + (1 - p_0)$$

- Tempo médio de espera na fila:

$$E[w] = \frac{[En_w]}{\lambda(K - E[n])}$$

- Tempo médio de resposta:

$$E[s] = E[w] + \frac{1}{\mu}$$

- Taxa efetiva de chegada:

$$\lambda' = \mu(1 - p_0)$$

**Exemplo**

Um sistema de uma empresa possui apenas um servidor. A taxa média de chegadas de requisições no sistema banco de dados é de quatro requisições/hora e o servidor processa em média 10 requisições por hora. Sendo o número total de funcionários da empresa igual a 10, calcule o número médio de requisições na fila do sistema.

Valores fornecidos no problema:

$m = 1$
$\lambda = 4$ requisições/hora
$\mu = 10$ pacotes/hora
$K = 10$ funcionários

Calculando o número médio de requisições na fila, tem-se

$$p_0 = \frac{1}{\sum_{n=0}^{10} \left(\frac{4}{10}\right)^n \frac{10!}{(10-n)!}} = 0{,}0002$$

e

$$E[n_w] = 10 - \left(\frac{4-10}{4}\right)(1 - 0{,}0002) = 11{,}5 \text{ requisiçõe}$$

O número médio de requisições na fila é de 11,5 requisições.

## 2.5.7 Fila M/M/∞//K – População Finita com Infinitos Servidores

Este tipo de sistema também possui população finita, mas com a diferença de ter servidores infinitos, isto é, cada requisição que chegar ao sistema terá um servidor pronto para atendê-la. A Figura 2.16 mostra a visualização da fila.

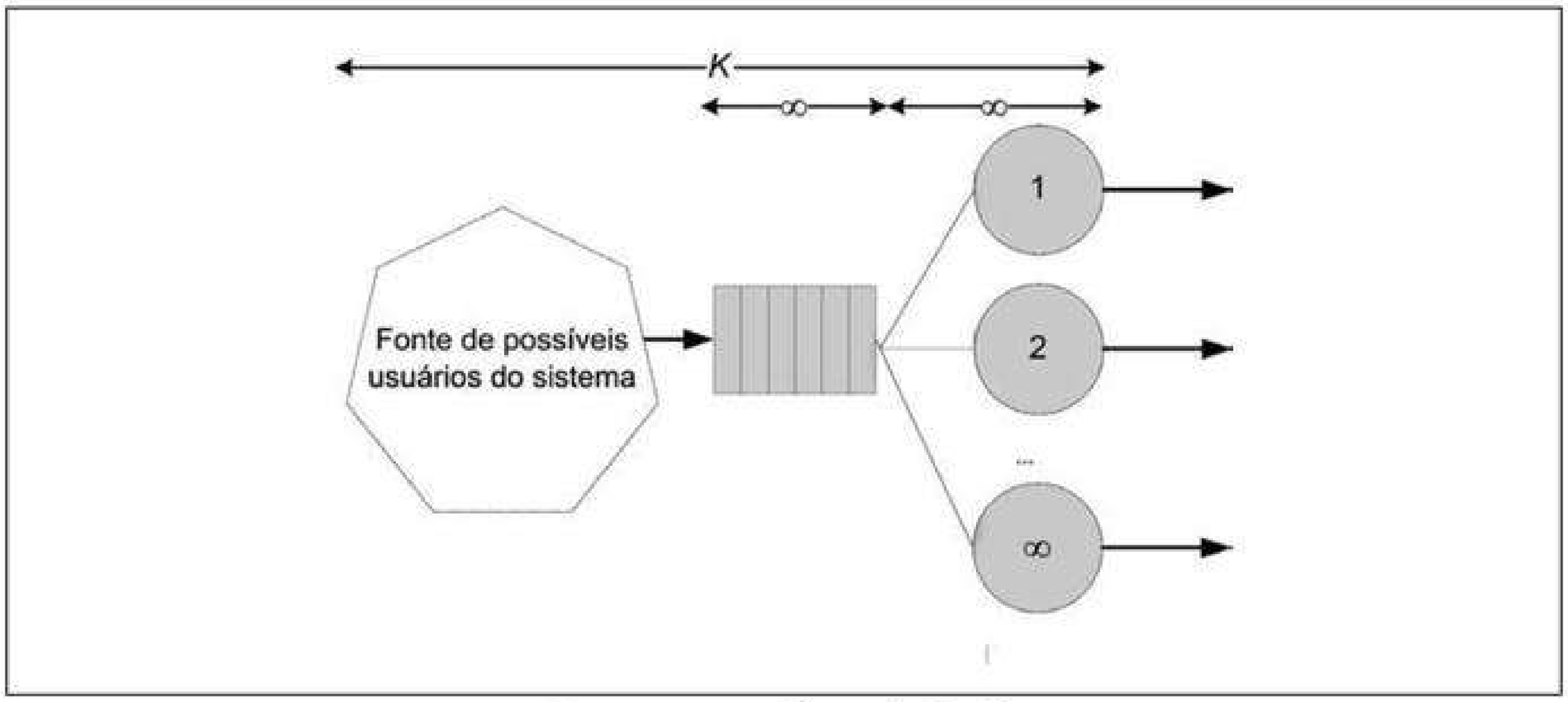


Figura 2.16 Fila M/M/∞//K.

A taxa de serviço depende do número de servidores ocupados e a taxa de chegadas depende do número de requisições já recebidas e sendo processadas ou na fila, até o limite de $K$ requisições da população:

$$\lambda_n = \begin{cases} \lambda(K-n), & 0 \le n \le K \\ 0, & n > K \end{cases}$$

$$\mu_n = n\mu, \quad n = 1, 2, \ldots$$

A Figura 2.17 mostra o diagrama de transição de estados para esta fila.

Do diagrama retiramos as probabilidades de $p_n$:

$$p_n = \begin{cases} \dfrac{\left(\dfrac{\lambda}{\mu}\right)^n \dbinom{K}{n}}{\left(1+\dfrac{\lambda}{\mu}\right)^K}, & 0 \le n \le K \\ 0, & n > K \end{cases}$$

e

$$p_0 = \frac{1}{\left(1+\dfrac{\lambda}{\mu}\right)^K}$$

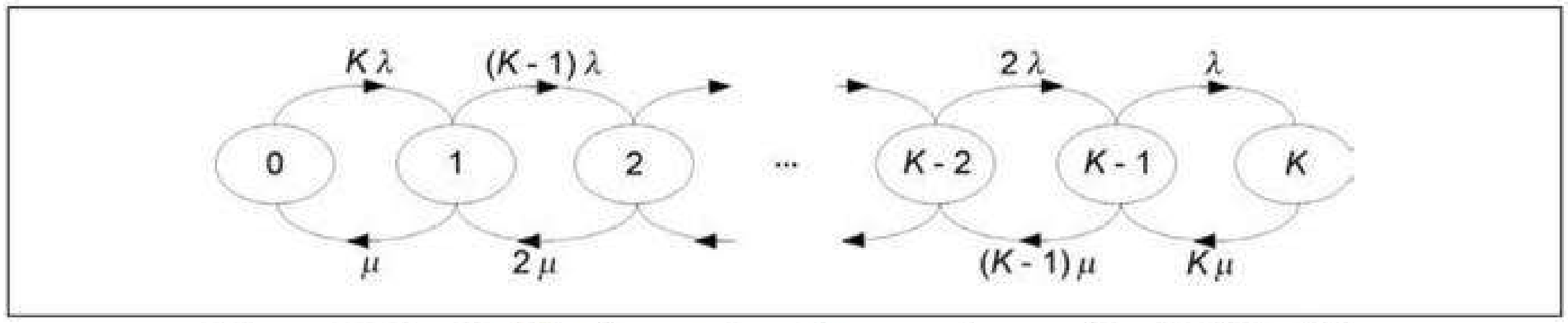


**Figura 2.17** Modelo de nascimento e morte para filas M/M/∞//K.

## Outras propriedades da fila M/M/∞//K

- Intensidade de tráfego:

$$\rho = \frac{K\lambda}{1+\frac{\lambda}{\mu}}$$

- Número médio de clientes no sistema:

$$E[n] = \frac{K\frac{\lambda}{\mu}}{1+\frac{\lambda}{\mu}}$$

**Exemplo**

Um sistema hipotético de bancos de dados possui infinitos servidores. A taxa média de chegadas de requisições em um sistema banco de dados é de quatro requisições/hora. O tempo médio para cada servidor completar o serviço do cliente é de seis minutos. Sendo o número total de clientes do banco de dados igual a 10, calcule o número médio de clientes no banco.

Valores fornecidos no problema:
$\lambda = 4$ clientes/hora
$\mu = 1/6$ minuto = 10 clientes/hora
$K = 1000$
$M$= infinitos servidores

O número médio de clientes no sistema de banco de dados é:

$$E[n] = \frac{K\frac{\lambda}{\mu}}{1+\frac{\lambda}{\mu}} = \frac{1000\frac{4}{10}}{1+\frac{4}{10}} = 384,62 \text{ clientes}$$

### 2.5.8 Fila M/M/m/B/K – m-Servidores com Capacidade e População Finitos

Este sistema tem o número de servidores limitados a $m$, capacidade do sistema (filas + servidores) limitada a receber $B$ clientes, e uma população finita $K$. É assumido que $K \geq B \geq m$ (Figura 2.18).

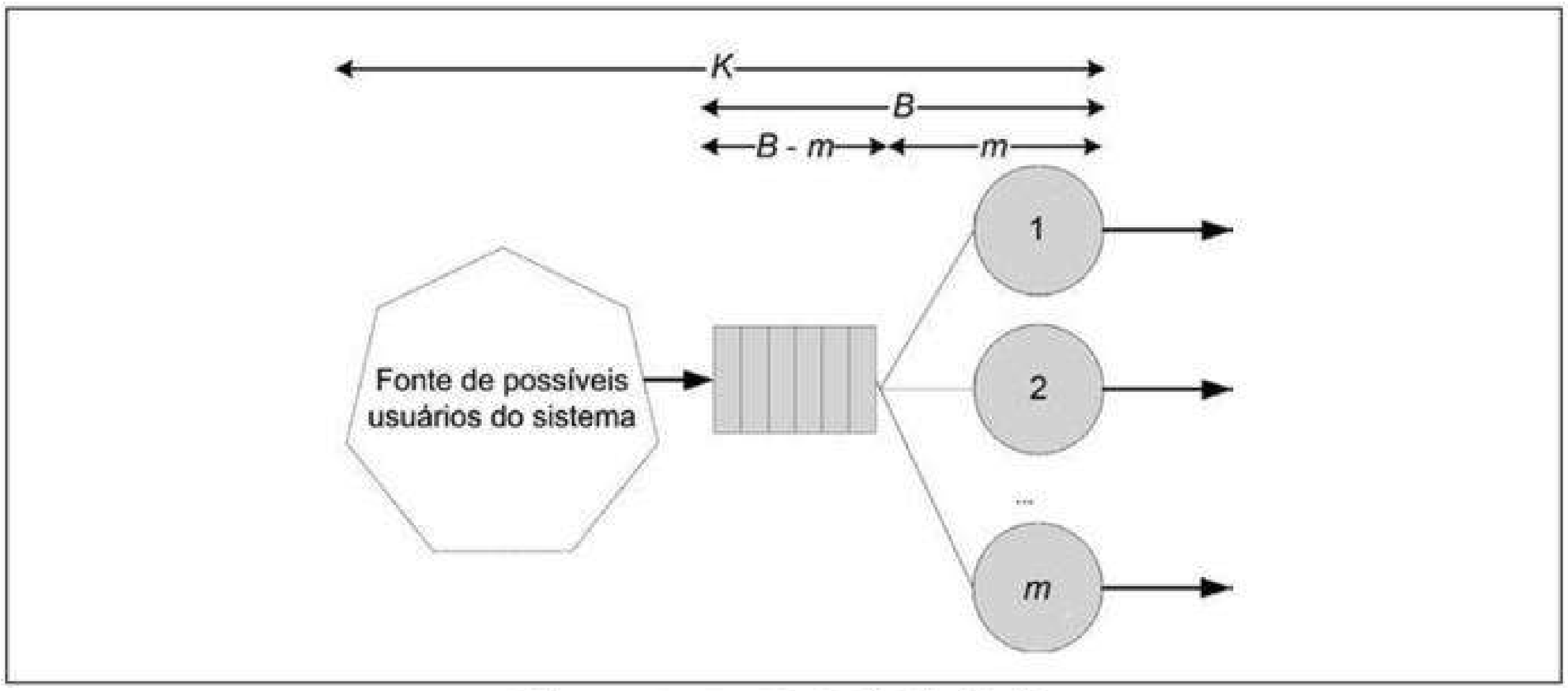


**Figura 2.18** Fila M/M/m/B/K.

Os clientes que chegarem ao sistema e encontrarem $B$ outros clientes já no sistema (capacidade total lotada) não poderão ser atendidos e serão perdidos, devendo 'chegar' novamente no sistema imediatamente. A taxa de serviço depende do número de servidores ocupados e a taxa de chegadas depende do número de requisições já recebidas até o limite da capacidade do sistema:

$$\lambda_n = \begin{cases} \lambda(K-n), & 0 \leq n \leq B-1 \\ 0, & n > B \end{cases}$$

$$\mu_n = \begin{cases} n\mu, & 0 \leq n \leq m \\ m\mu, & n \geq m \end{cases}$$

A Figura 2.19 mostra o diagrama de transição de estados para esta fila.

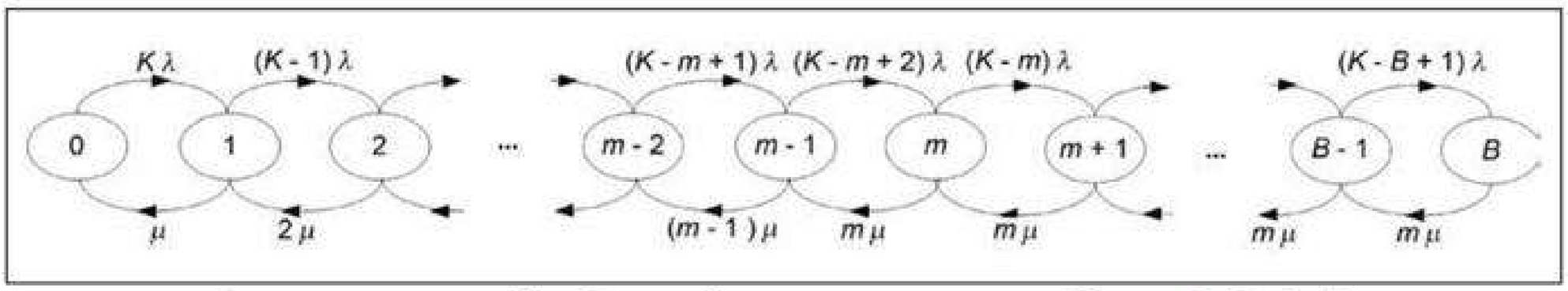


**Figura 2.19** Modelo de nascimento e morte para filas M/M/m/B/K.

Do diagrama retiramos as probabilidades de $p_n$:

$$p_n = \begin{cases} \left(\frac{\lambda}{\mu}\right)^n \binom{K}{n} p_0, & 0 \le n \le m-1 \\ \left(\frac{\lambda}{\mu}\right)^n \binom{K}{n} \frac{n!}{m!} m^{m-n} p_0, & m \le n \le K \end{cases}$$

## Outras propriedades da fila M/M/m/B/K

- Intensidade de tráfego:

$$\rho = \lambda / m\mu$$

- Utilização:

$$U = \rho(K - E[n] - (K - B)p_B)$$

- Número médio de usuários no sistema:

$$E[n] = \sum_{n=0}^{m} n \cdot p_n$$

- Tempo médio de espera na fila:

$$E[w] = \frac{[En]}{\lambda'}$$

- Tempo médio de resposta:

$$E[s] = \frac{E[n]}{\lambda(K - E[n] - (K - B)pB)}$$

- Taxa efetiva de chegadas:

$$\lambda' = \lambda(N - E[n])$$

**Exemplo**

Um serviço Web de reserva de automóveis possui dois servidores. O processo de chegada de requisições de reserva obedece a uma distribuição de Poisson, com taxa média de quatro requisições por hora. O tempo de processamento de instruções segue uma distribuição exponencial com média de dois minutos. No serviço com espera, isto é, quando dois servidores estiverem ocupados, as chamadas que chegam ficam em um buffer com capacidade de quatro requisições.

*(continua)*

Sendo o número máximo de clientes que usam a rede igual a 20, e a probabilidade de o sistema estar vazio de 11%, determine qual a probabilidade de a rede ter quatro clientes querendo realizar reservas.

Valores fornecidos no problema:
$\lambda = 8$ requisições/hora
$\mu = 1/2$ minuto = 30 requisições por hora
$m = 2$ servidores
$B = 6$ (2 requisições sendo servidas e 4 vagas na fila)
$K = 20$
$p_0 = 11\% = 0{,}11$
Calculando $p_4$, com $m \leq 4 \leq K$, tem-se:

$$p_4 = \left(\frac{\lambda}{\mu}\right)^n \binom{K}{n} p_0 = \left(\frac{6}{30}\right)^4 \left(\frac{20!}{4!(20-4!)}\right) 0{,}11 = 0{,}85$$

A probabilidade de existirem quatro clientes no sistema é de 85%.

## 2.6 Exercícios

1) Numa sub-rede IP, os pacotes que chegam a um determinado nó da rede são processados por um único processador. Os pacotes chegam a este nó de acordo com o processo de Poisson, com uma taxa média de cinco pacotes por segundo. Além disso, o tempo de processamento pode ser caracterizado por uma variável aleatória exponencial de média igual a 0,1 segundo. Sabe-se que quando o processador está ocupado os pacotes que chegam ao nó aguardam até que possam ser processados, sem limite de armazenamento.

   a) Determine o atraso médio experimentado pelos pacotes ao passarem pelo nó em consideração.
   b) Um processador foi adicionado ao nó. Determine o novo valor do atraso médio experimentado pelos pacotes.
   c) Determine o número médio de pacotes que estão esperando para serem processados em certo momento.

2) Um servidor de uma aplicação recebe requisições de clientes de acordo com o processo de Poisson, a uma taxa de chegada média de 30 requisições por hora. O tempo de serviço por requisição demora em média 1,5 min. Determine:

   a) O número médio de requisições que esperam para serem atendidas na fila.
   b) O tempo que uma requisição espera para ser atendida.
   c) A probabilidade que o servidor esteja desocupado.

3) Uma central de encaminhamento de chamadas telefônicas tem cinco ramais para uso dos funcionários de uma empresa. Durante a tarde, os funcionários que chamadas enviam solicitações que chegam à central de acordo com o processo de Poisson, a uma taxa de 100 por hora. A duração média de uma chamada é de dois minutos. Determine:

   a) O número médio de solicitações, usando ou esperando por um ramal;
   b) O tempo médio de espera de uma solicitação ser atendida.

4) Um *firewall* tem duas portas; uma para filtros de dados de entrada e outra para filtro de saída em uma rede IP. O tempo de serviço para cada porta é distribuído exponencialmente, com uma média de um segundo. Os pacotes chegam ao equipamento de acordo com o processo de Poisson, com taxa média de 40 por minuto. Os pacotes que chegam ao *firewall* para entrar ou sair da rede seguem o processo de Poisson próprio, com uma taxa média de 20 por minuto. O administrador da rede deseja modificar o sistema para permitir que cada porta possa atender a entradas e saídas de pacotes. Espera-se que o tempo de serviço médio de cada porta aumente para 1,2 segundo, e que a fila de pacotes em cada porta diminua. Analise as duas opções de configuração do *firewall* em relação ao tempo sem atividade e o número esperado de pacotes em qualquer momento determinado.

5) Consultas SQL chegam a um servidor de banco de dados de acordo com o processo de Poisson com média de cinco consultas por segundo. O tempo de atendimento de cada consulta é em média de 10 segundos. A fila do servidor recebe apenas quatro requisições e as consultas que não conseguem lugar são reencaminhadas para outro serviço.

   a) Calcule o número médio de consultas que vão embora sem serviço durante uma hora.
   b) Qual é o tempo médio de permanência de uma consulta no sistema?
   c) Qual é a probabilidade de uma consulta ser atendida assim que chega?

# Análise Operacional de Fila Isolada

3

## 3.1 Introdução

O uso de filas para caracterização de sistemas computacionais e cálculo de desempenho pode ser utilizado pela Teoria da Análise Operacional, que é uma teoria para formulação de modelos analíticos de sistemas computacionais e se baseia no relacionamento entre quantidades medidas durante um intervalo de observação do sistema.

A análise operacional de uma fila única (o mesmo tipo de fila analisado no Capítulo 2) considera os mesmos parâmetros de caracterização de fila, como chegadas de requisições, tempo de serviço, tempo de resposta, utilização, entre outros.

As quantidades medidas, como número de clientes sendo atendidos no sistema, podem ser contabilizadas através de monitores de software ou hardware para uso nos cálculos de desempenho. As quantidades usadas nos cálculos são chamadas de variáveis operacionais e classificadas em dois tipos:

- Variáveis operacionais básicas, medidas diretamente no sistema. Por exemplo, número de operações de consulta completadas.
- Variáveis operacionais derivadas, obtidas através das relações entre variáveis operacionais básicas, usando as leis operacionais que serão vistas neste capítulo. Por exemplo, obtenção do tempo de resposta do sistema de consultas.

Neste capítulo serão abordados os conceitos de análise operacional para filas isoladas, e no próximo capítulo será enfocada a análise operacional para sistemas de filas em rede.

## 3.2 Tipos de Centro de Serviço

Um sistema de fila é caracterizado pela forma como atende as requisições que chegam. O centro de serviço (ou servidor, ou recurso) pode atender às requisições independentemente do tipo de carga recebido. Pode ser dependente ou independente do tipo de carga, ou ser 'infinito'.

### Independente de carga

São os centros de serviços que possuem taxa de serviço constante, isto é, que não depende da carga recebida. Por exemplo, o acesso a um dispositivo de E/S.

A Figura 3.1 mostra a representação de uma fila simples (a mesma representação vista no estudo de Teoria das Filas, Capítulo 2).

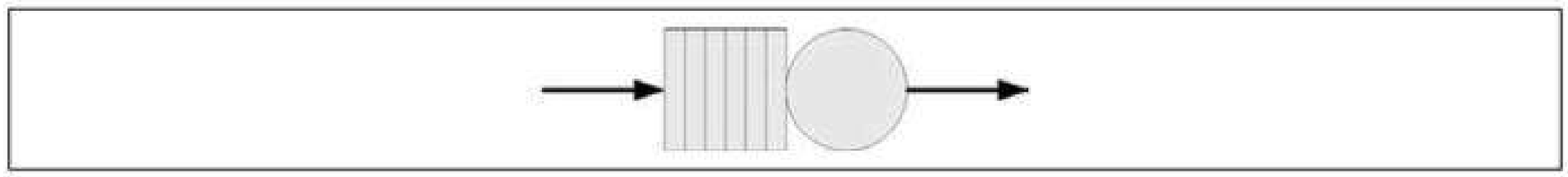

**Figura 3.1** Representação de recurso independente de carga.

## Dependente de carga

São os centros de serviços cuja taxa de serviço (e tempo de serviço) depende do número de clientes na fila. Por exemplo, elementos de um sistema computacional que pode modificar seu comportamento de acordo com o número de usuários acessando-o, como um disco rígido com políticas diferenciadas de atendimento de clientes.

A Figura 3.2 mostra a representação de uma fila dependente de carga.

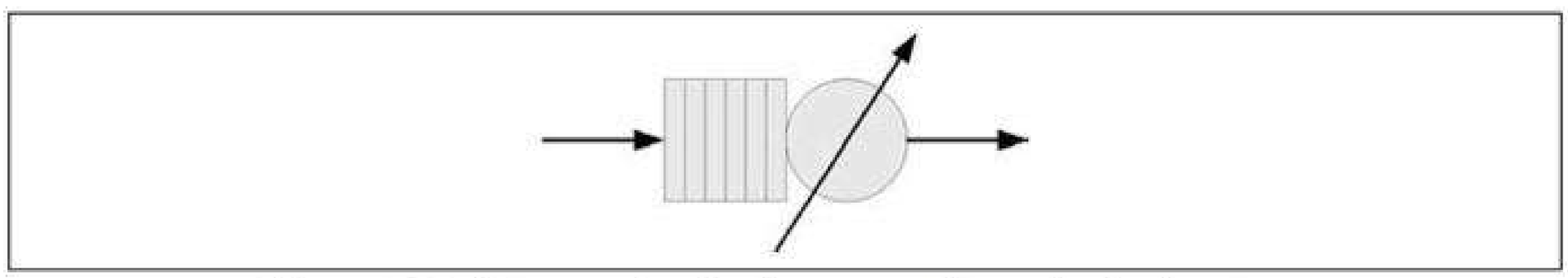

**Figura 3.2** Representação de recurso dependente de carga.

## Centros de atrasos

Os centros de atraso não geram enfileiramento de requisições/clientes, isto é, nunca geram fila. Cada nova requisição que chega ao sistema tem um novo centro de serviço alocado para atender o serviço, portanto não ocorre competição por um mesmo centro de serviço. Por exemplo, pode-se usar um centro de atraso para representar a latência de uma linha de transmissão de dados.

A Figura 3.3 mostra a representação de um centro de atraso.

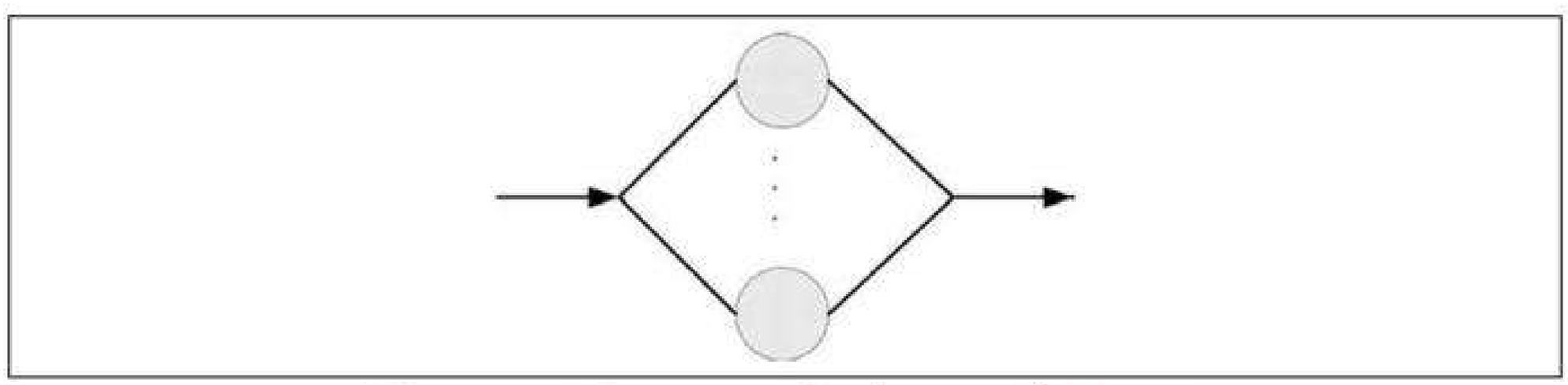

**Figura 3.3** Representação de centro de atraso.

Portanto, o tempo de residência de um cliente (ou requisição) no centro de atraso é exatamente a demanda de serviço do cliente.

Os centros de atraso também são úteis para representar situações onde existe um atraso conhecido. Por exemplo, pode-se usar esse tipo de recurso para representar a latência de uma transmissão de dados via satélite.

## 3.3 Variáveis Operacionais

As variáveis operacionais básicas são:

- ***A*** número de requisições que chegam à fila no intervalo de observação.
- ***B*** tempo ocupado, ou seja, o tempo que o servidor possui requisições em atendimento.
- ***C*** número de requisições atendidas pelo servidor durante certo intervalo de tempo.
- ***S*** tempo de serviço, isto é, tempo que o servidor gasta para atender a uma requisição.

$$S = \frac{B}{C}$$

- **λ** taxa de média de chegadas de requisições no sistema.

$$\lambda = \frac{A}{T}$$

- ***X*** taxa de processamento ou vazão, ou seja, o número de requisições atendidas por unidade de tempo.

$$X = \frac{C}{T}$$

- ***U*** utilização do servidor.

$$U = \frac{B}{T}$$

A Figura 3.4 mostra a representação das variáveis operacionais em um sistema de fila única.

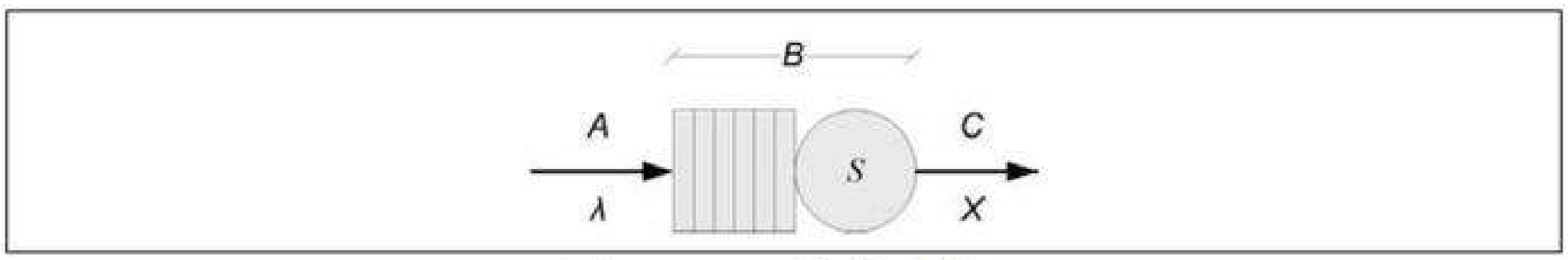


**Figura 3.4** Fila isolada.

**Exemplo**

Um servidor de arquivos está conectado à rede da empresa. Ele recebe 50 requisições de arquivos e atende a 20 dessas requisições durante um intervalo de observação de 10 minutos. O servidor gasta 10 segundos para atender a cada requisição. Os valores das variáveis operacionais do servidor de arquivos são:

$A = 50$ requisições
$B = 3$ minutos $= 3 \times 60 = 180$ segundos
$C = 20$ requisições

$$S = \frac{B}{C} = \frac{180}{20} = 90 \text{ segundos}$$

$$T_{\text{observado}} = 10 \text{ minutos} = 10 \times 60 = 600 \text{ segundos}$$

$$\lambda = \frac{A}{T} = \frac{50}{600} = 0{,}083 \text{ requisição / segundo}$$

$$X = \frac{C}{T} = \frac{20}{600} = 0{,}033 \text{ requisição / segundo}$$

$$U = \frac{B}{T} = \frac{180}{600} = 0{,}3 = 30\%$$

## 3.4 Leis Fundamentais e Teoremas Operacionais

Algumas leis foram criadas para facilitar a análise de desempenho de sistemas de fila. As leis fundamentais de avaliação são a Lei da Utilização, Lei de Little e Lei do Tempo de Resposta Interativo. Também existe a Hipótese do Equilíbrio de Fluxo e os seguintes teoremas: Teorema da Taxa de Processamento e o Teorema do Tempo de Resposta.

### *3.4.1 Lei da Utilização*

Dado um número de clientes atendidos $C$, um tempo de ocupação $B$, de um sistema de fila, durante um intervalo de observação $T$, a seguinte relação para cálculo da utilização $(U)$ do sistema é válida:

$$U = \frac{B}{T} = \frac{C}{T} \times \frac{B}{C} = XS$$

A utilização do sistema não pode ser maior do que 100% $(U \leq 1)$, logo $X \leq 1/S$.

**Exemplo 1**

Um segmento de rede recebe 5000 pacotes por segundo para transmissão. Cada pacote tem um tempo de transmissão igual a 1,5 milissegundo. Qual é a utilização do segmento de rede?

Valores fornecidos no problema:
$\lambda = 500$ pacotes/segundo
$S = 1{,}5$ ms $= 0{,}0015$ segundo
Pela Lei da Utilização, tem-se:

$$U = XS = 500 \times 0{,}0015 = 0{,}75 = 75\%$$

A utilização do sistema é de 75%.

**Exemplo 2**

Durante um período de observação de oito segundos, 400 consultas foram atendidas por um servidor de banco de dados que pode atender 250 consultas por segundo. A utilização do servidor aumenta se o número de consultas atendidas for de 900 consultas?

Valores fornecidos no problema:

$T_{\text{observado}} = 8$ segundos

$C_{\text{inicial}} = 400$ consultas

$S = 1/250 = 0{,}004$ segundo (o tempo de serviço é o inverso da taxa de serviço, como visto no Capítulo 2)

$$X_{\text{inicial}} = \frac{C}{T} = \frac{400}{8} = 50 \text{ consultas / segundo}$$

Pela Lei da Utilização, tem-se

$$U = XS = 50 \times 0{,}004 = 0{,}2 = 20\%$$

$$C_{\text{final}} = 900 \text{ consultas}$$

$$X_{\text{final}} = \frac{C}{T} = \frac{900}{8} = 112{,}5 \text{ consultas / segundo}$$

Pela Lei da Utilização, tem-se

$$U = XS = 112{,}5 \times 0{,}004 = 0{,}45 = 45\%$$

*(continua)*

Portanto,

$$\text{Variação}_{\text{utilização}} = \frac{\text{Utilização}_{\text{final}}}{\text{Utilização}_{\text{inicial}}} = \frac{0{,}45}{0{,}2} = 2{,}25$$

A utilização final é 2,5 maior do que a utilização inicial.

### 3.4.2 Lei de Little

A Lei de Little diz que o número médio de clientes no sistema ($N$) é igual à taxa de chegadas no sistema ($\lambda$) vezes o tempo médio gasto no sistema (tempo de resposta $R$):

$$N = \lambda R$$

Também pode ser usado para cálculo do número médio de clientes na fila ($N_q$), que é equivalente à taxa de chegadas multiplicada pelo tempo médio de serviço:

$$N_q = \lambda S$$

**Exemplo 1**

Um roteador possui tempo médio total para atender um pedido de encaminhamento de pacotes de 30 milissegundos. A taxa de chegadas de pacotes é de 500 pacotes por segundo. Qual o número médio de pacotes no roteador?

Valores fornecidos no problema:

$\lambda$ = 500 pacotes/segundo

$R$ = 30 ms = 0,03 segundo

Pela Lei de Little tem-se:

$$N = \lambda R = 500 \times 0{,}03 = 15 \text{ pacotes}$$

O número médio de pacotes no roteador é 15.

**Exemplo 2**

Um servidor de arquivos foi monitorado por 30 minutos e o número de requisições para operação de E/S recebidas pela rede da empresa nesse período foi de 4600. O número ativo de requisições de arquivos era de três. Qual o tempo médio de resposta por requisição no servidor?

Valores fornecidos no problema:

*(continua)*

$T_{\text{observado}}$ = 30 minutos = 30 × 10 = 300 segundos

$C$ = 4600 requisições

$N$ = 3 requisições

$$\lambda = \frac{A}{T} = \frac{4600}{300} = 15{,}33 \text{ requisições/segundo}$$

Pela Lei de Little, tem-se:

$$N = \lambda R \Rightarrow R = \frac{N}{\lambda} = \frac{3}{15{,}33} = 0{,}20 \text{ s}$$

O tempo de resposta do servidor de arquivos é de 0,20 segundo.

### 3.4.3 Lei do Tempo de Resposta Interativo

Um sistema interativo é caracterizado por possuir um número fixo de clientes conectados a um sistema central, para onde suas requisições são enviadas e processadas. A requisição atendida é devolvida ao cliente com a resposta do processamento requisitado. Depois do recebimento de uma requisição atendida, o cliente fica um tempo parado, chamado de *tempo de pensar*, que seria o tempo no qual o cliente está pensando ou escrevendo a sua nova solicitação ao subsistema central. O subsistema tem uma vazão das requisições que foram atendidas e devolvidas aos clientes.

A Figura 3.5 mostra um exemplo de um sistema interativo, composto de m terminais de usuários que acessam um subsistema central.

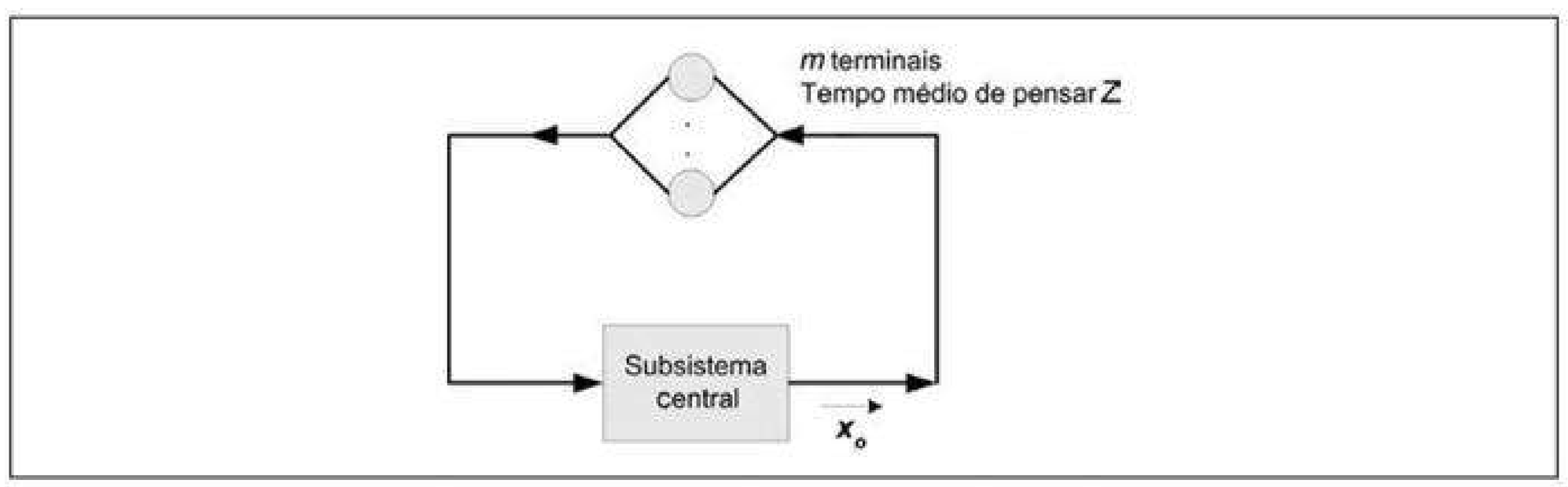


**Figura 3.5** Exemplo de sistema interativo.

O tempo de resposta ($R$) de um sistema interativo é dado por:

$$R = \frac{N}{X_o} - Z$$

Onde $X_o$ é a vazão média (requisições atendidas por unidade de tempo), $N$ número de usuários dentro do subsistema central e $Z$ o tempo de pensar.

**Exemplo 1**

Um sistema de computação interativo foi observado durante um período no qual 36 terminais estavam ativos. Durante esse intervalo, a taxa de processamento foi de quatro transações por segundo e o tempo médio de pensar foi de seis segundos. Qual o tempo médio de resposta?

Valores fornecidos no problema:

$N = 36$ usuários

$X = 4$ transações por segundo

$Z = 6$ segundos

Utilizando a Lei do Tempo de Resposta Interativo, tem-se:

$$R = \frac{N}{X_o} - Z = \frac{36}{4} - 6 = 3 \text{ s}$$

O tempo de resposta do sistema é de 3 segundos.

**Exemplo 2**

Usando como base o sistema representado na Figura 3.5, uma requisição de usuário submetida ao sistema deve ficar na fila por memória, e pode começar a processar (no subsistema central) somente quando conseguir uma partição de memória. Se existem 100 usuários ativos com tempo de pensar de 20 s, e o tempo de resposta do sistema (soma da fila na memória e tempo na CPU) é de 10 s, quantos clientes estão competindo por memória?

Valores fornecidos no problema para o sistema interativo:

$N = 100$ usuários

$R = 10$ segundos

$Z = 20$ segundos

Utilizando a Lei do Tempo de Resposta Interativo, tem-se:

$$R = \frac{N}{X_o} - Z \Rightarrow X_o = \frac{N}{(R+Z)} = \frac{100}{10+20} = 3{,}33 \text{ usuários/segundo}$$

Para calcular quantos usuários estão competindo por memória no subsistema (excluindo os usuários que estão 'pensando'), é possível usar a Lei de Little, isolando o subsistema central:

$$N_{\text{memória}} = X_o R = 3{,}33 \times 20 = 66{,}6 \text{ usuários}$$

### 3.4.4 Hipótese do Equilíbrio de Fluxo

Esta hipótese diz que "o estado inicial de uma fila isolada é igual ao seu estado final". Podemos interpretá-la de duas formas:

- O número de transações na fila no instante inicial $(N(0))$ é igual ao número observado no instante final $(N(T))$:

$$N(0) = N(T)$$

- O número de chegadas deve ser igual ao número de saídas durante o intervalo de observação, ou a taxa de chegadas de clientes no sistema deve ser igual à vazão (Figura 3.6):

$$\lambda = X$$

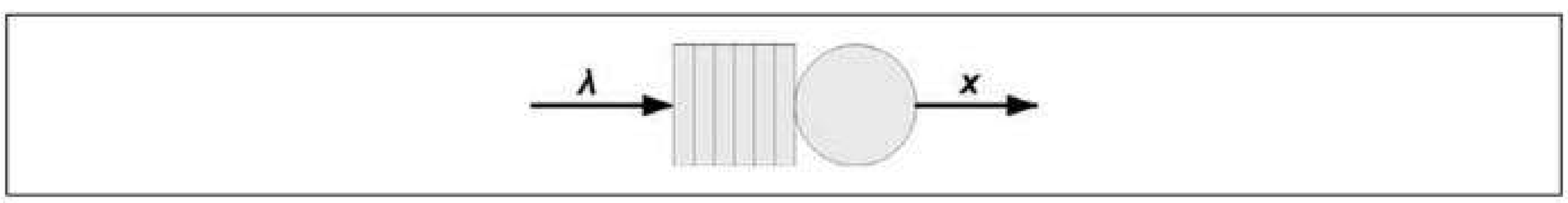


**Figura 3.6** Equilíbrio de fluxo de taxas.

Utilizando esta hipótese na Lei de Little, tem-se então que:

$$N = XR$$

e

$$N_q = XS$$

**Exemplo**

A vazão de um sistema de compartilhamento de terminais foi medida em 2400 requisições atendidas durante um período de observação de 10 minutos. Se o número médio de requisições no sistema foi de quatro durante esse período, qual o tempo médio de resposta do sistema?
Valores fornecidos no problema:

$C =$ 2400 requisições

$T_{\text{observado}} = 10 \text{ min} = 10 \times 60 = 600$ segundos

$N =$ 4 requisições

*(continua)*

Podemos calcular a vazão $X$:

$$X = \frac{C}{T} = \frac{2400}{600} = 4 \text{ requisições/segundo}$$

Usando a Lei de Little, tem-se

$$N = XR \Rightarrow R = \frac{N}{X} = \frac{4}{4} = 1\,\text{s}$$

O tempo de resposta do sistema é de 1 segundo.

### 3.4.5 Teorema da Taxa de Processamento

Utilizando a Hipótese de Equilíbrio de Fluxo, a partir da Lei da Utilização, tem-se que:

$$U = \lambda S$$

**Exemplo 1**

Durante um período de um minuto, 300 consultas SQL foram recebidas por um servidor de banco de dados. O tempo médio de atendimento para cada consulta é de 30 milissegundos. Calcule a utilização média do servidor neste período.

Valores fornecidos no problema:

$C$ = 300 consultas

$T$ = 1 min= 1× 60 = 60 segundos

$S$ = 30 ms = 0,03 segundo

Pela Hipótese de Equilíbrio de Fluxo:

$$\lambda = X = \frac{C}{T} = \frac{300}{60} = 5 \text{ consultas/segundo}$$

Calculando a utilização, tem-se:

$$U = XS = 5 \times 0{,}03 = 0{,}15 = 15\%$$

A utilização do sistema foi de 15%.

**Exemplo 2**

Um servidor Web possui tempo de resposta para atender requisições HTTP de dois milissegundos. Considerando diversas taxas de chegadas (10, 100 e 300 requisições por segundo), qual a utilização do servidor para as diversas taxas?

*(continua)*

Valores fornecidos no problema:

$S$ = 2 ms = 0,002 segundo

$\lambda_1$ = 10 requisições/segundo

$\lambda_2$ = 100 requisições/segundo

$\lambda_3$ = 300 requisições/segundo

Calculando a utilização, tem-se:

$$U_1 = \lambda_1 S \; 10 \times 0{,}002 = 0{,}02 = 0{,}2\%$$

$$U_2 = \lambda_2 S \; 100 \times 0{,}002 = 0{,}2 = 20\%$$

$$U_3 = \lambda_3 S \; 300 \times 0{,}002 = 0{,}6 = 60\%$$

### *3.4.6 Teorema do Tempo de Resposta*

Neste teorema, o tempo médio de resposta de uma transação na fila isolada pode ser calculado em função da utilização e do tempo médio de serviço. Segundo este teorema, o tempo de resposta é dado por:

$$R = \frac{S}{1-U}$$

Por este teorema podemos observar que o tempo de resposta do sistema tende para infinito à medida que a utilização do sistema aumenta até 100%.

Combinando o Teorema da Taxa de Processamento com a Lei de Little tem-se que:

$$N = \lambda R = \frac{\lambda S}{1-U} = \frac{U}{1-U}$$

**Exemplo 1**

O tempo médio de serviço de um disco é de 20 milissegundos e ele recebe em média 30 pedidos de E/S por segundo. Qual o tempo de resposta do disco?

Valores fornecidos no problema:

$S$ = 20 ms = 0,02 segundo

$\lambda$ = 30 pedidos/segundo

Pelo Teorema da Taxa de Processamento, tem-se:

$$U = \lambda S = 30 \times 0{,}02 = 0{,}6 = 60\%$$

Pelo Teorema do Tempo de Resposta, tem-se:

$$R = \frac{S}{1-U} = \frac{0{,}02}{1-0{,}6} = 0{,}05\ s$$

O tempo médio de resposta do disco é de 0,05 segundo.

**Exemplo 2**

Uma rede local opera com 100 computadores e um servidor, que mantém um banco de dados para consulta. O tempo médio para o servidor processar e responder cada consulta é de 0,5 segundo e, no horário de pico, são recebidas 40 consultas por minuto.

a) Qual o tempo médio de resposta do servidor?

Valores fornecidos no problema:

$S = 0{,}6$ segundo

$\lambda = 40$ consultas/minuto $= 40/60 = 0{,}67$ consulta/segundo

Pelo Teorema da Taxa de Processamento, tem-se:

$$U = \lambda S = 0{,}67 \times 0{,}6 = 0{,}402 = 40{,}2\%$$

Pelo Teorema do Tempo de Resposta, tem-se:

$$R = \frac{S}{1-U} = \frac{0{,}6}{1-402} = 1\,\text{s}$$

O tempo médio de resposta do disco é de aproximadamente 1 segundo.

b) Se a carga crescer em 20%, qual a variação do tempo de resposta?

$\lambda_2 = 40$ consultas/min + 20% $= 40 \times 1{,}2 = 48$ consultas/min $= 0{,}8$ consulta/s

$$U_2 = \lambda_2 S = 0{,}8 \times 0{,}6 = 0{,}48 = 48\%$$

$$R_2 = \frac{S}{1-U_2} = \frac{0{,}6}{1-48} = 1{,}25\,\text{s}$$

$$\text{Variação}_R = \frac{R_2}{R} = \frac{1{,}25}{1} = 1{,}25$$

## 3.5 Exercícios

1) Um segmento de rede transmite 1000 pacotes por segundo. Cada pacote tem um tempo médio de transmissão igual a 10 milissegundos. Qual é a utilização do segmento de LAN?

2) O roteador da Figura 3.7 tem uma latência de 200 milissegundos por pacote. Pacotes saem da LAN B para a LAN C numa taxa de 30.000 pacotes por segundo e da LAN A para a LAN B a uma taxa de 20.000 pacotes por segundo. Qual é o número médio de pacotes no roteador em trânsito da LAN B para a LAN C e da LAN A para a LAN B?

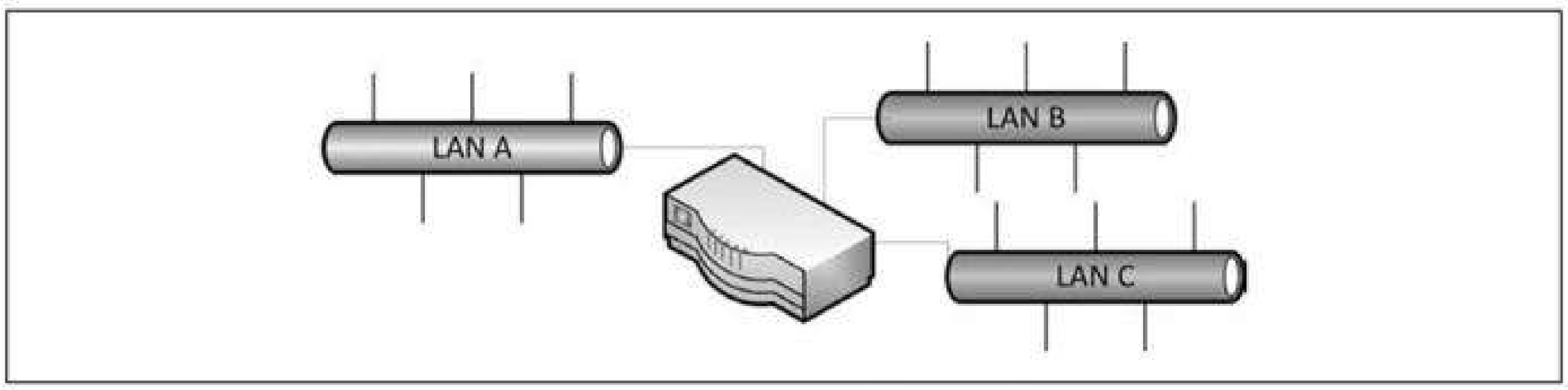


**Figura 3.7** Exemplo de roteador.

3) Um sistema de servidor de arquivos terá seu desempenho testado com a variação do número de requisições (10, 50 e 100 requisições por minuto). Sendo o tempo de serviço 20 milissegundos por requisição, qual é o número médio de requisições na fila nestes casos?

4) Considere um roteador que conecta uma rede local a uma rede de comutação de pacotes X.25, e analise o tráfego da rede através do roteador. Os pacotes chegam para ser transmitidos com uma média de cinco por segundo. O tamanho do pacote é de 40 bytes. A linha que conecta o roteador com a rede X.25 possui a vazão de 9,6 Kbps. Calcule:

   a) Qual a média do tempo de fila no roteador?
   b) Quantas mensagens em média ficam no roteador, incluindo as que esperam pela transmissão e a que está sendo transmitida?

5) Durante um período de 10 segundos, 80 requisições foram atendidas por um servidor de arquivos. O tempo médio de serviço no disco é de 60 milissegundos. Qual é a utilização média do disco do servidor nesse período?

6) Continuando o Exemplo 2 do tópico Lei do Tempo de Resposta Interativo, calcule: se o tempo na fila por memória foi de 8 s, qual o número médio de clientes carregados na memória?

# Análise Operacional de Redes de Filas

4

## 4.1 Introdução

Uma rede de filas é uma coleção de centros de serviços e clientes. Uma rede de filas caracteriza-se por ser um sistema que contém duas ou mais filas conectadas ou não entre si. As requisições chegam ao sistema por uma ou mais entradas do exterior, trafegam de uma fila para outra, e saem por uma ou mais saídas.

O caso de um sistema de uma fila é o sistema de fila única, visto no Capítulo 3.

A Figura 4.1 mostra um exemplo de rede de filas.

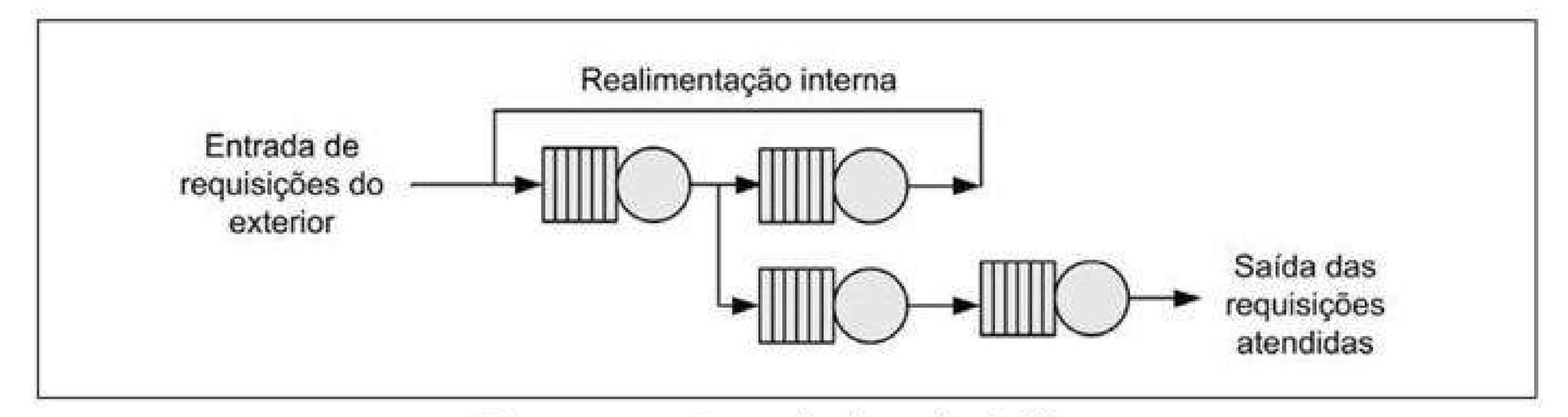


**Figura 4.1** Exemplo de rede de filas.

O sistema computacional é representado por uma rede de filas, que é avaliada analiticamente. A avaliação analítica envolve a solução de um conjunto de equações induzido pela rede de filas e seus parâmetros.

## 4.2 Classificação

Uma rede de filas pode ser classificada quanto a seu tipo, carga e número de cargas.

### 4.2.1 Tipos de Filas

#### Filas abertas

São aquelas que admitem chegadas de requisições do exterior e saídas das requisições processadas para o exterior. A Figura 4.2 mostra um exemplo de rede de filas contendo três filas em sequência.

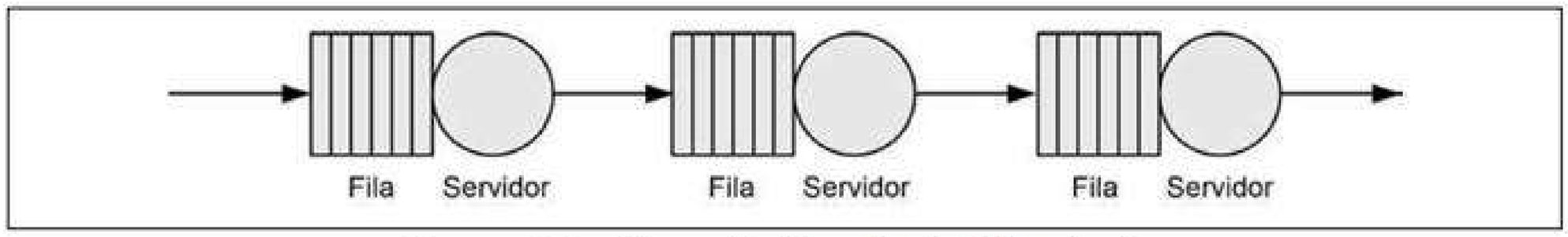


**Figura 4.2** Exemplo de rede de filas aberta.

#### Filas fechadas

Essas filas não admitem entradas nem saídas de requisições. Nelas, o número de requisições é sempre constante. A Figura 4.3 mostra uma rede de filas contendo três filas em sequência.

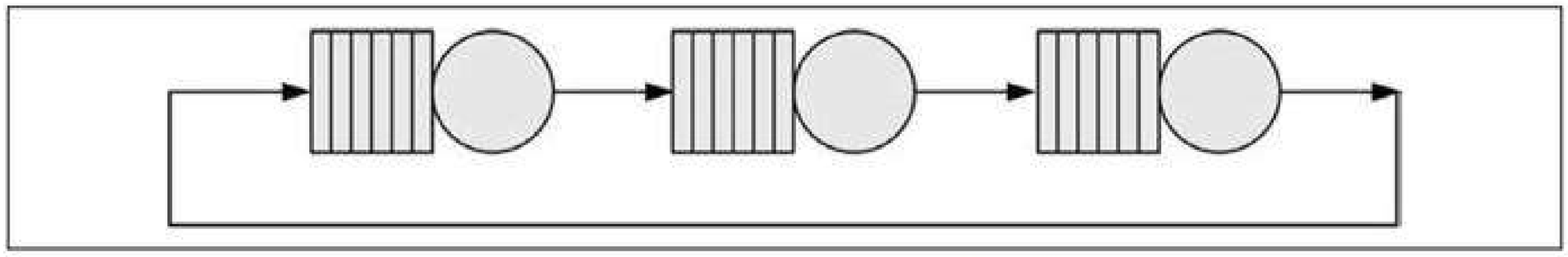

**Figura 4.3** Exemplo de rede de filas fechada.

#### Filas mistas

Filas mistas possuem partes abertas e fechadas ao mesmo tempo. Nesse caso, parte do sistema se comporta de forma fechada e parte do sistema se comporta de forma aberta.

A Figura 4.4 mostra o exemplo de uma rede de filas mista. Na figura, a parte da rede com traço mais fino é a parte fechada, enquanto o restante trafega de forma aberta.

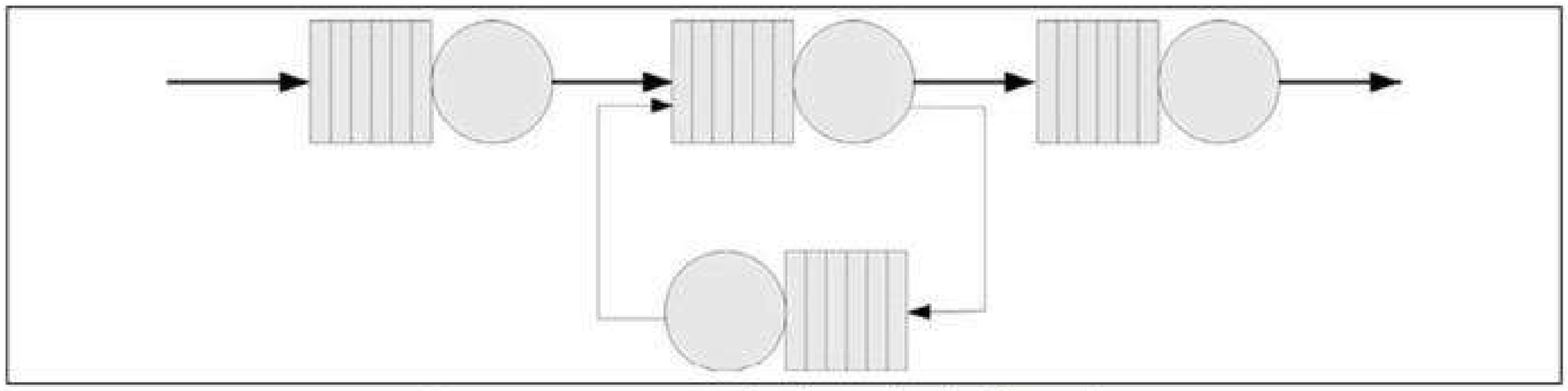

**Figura 4.4** Exemplo de rede de filas mista.

### 4.2.2 Tipos de Classes de Cargas

A carga de requisições ou clientes que trafega na rede pode ser classificada em aberta ou fechada.

#### Cargas abertas

Uma rede com carga aberta permite que o sistema receba um número ilimitado de clientes. Por exemplo, um roteador recebe infinitos pacotes para encaminhamento. Cargas de trabalho do tipo *transação* são especificadas pelo parâmetro de taxa de chegadas ($\lambda$) de novos clientes ao sistema. A população de clientes varia com o tempo e os clientes devem sair do sistema após o término do serviço. Modelos de filas com carga do tipo transação são chamados de *modelos abertos*.

#### Cargas fechadas

Um sistema de carga fechada permite apenas a entrada de um número de clientes no sistema limitado e conhecido. Por exemplo, cargas interativas e *batch*.

São dois os tipos de carga considerados fechados:

- Carga em lote (*batch*): a variável $N$ indica o número de clientes de trabalhos ativos no sistema, possui população finita, e os clientes com serviço terminado saem do sistema e são substituídos, instantaneamente, por outro grupo de trabalhos em espera.
- Carga de terminal: $N$ indica o número de clientes ativos e $Z$ indica o tempo de pensar.

Modelos de filas fechados usam cargas fechadas. Filas mistas usam em conjunto cargas fechadas e cargas abertas.

### 4.2.3 Número de Classes de Cargas

Uma rede de filas pode ter um só tipo de requisição (ou clientes) ou mesmo diversos tipos de requisições ao mesmo tempo. As cargas podem ser então do tipo único ou múltiplo.

#### Carga única

As requisições que trafegam na rede de filas são de um mesmo tipo. Todas as requisições possuem as mesmas características de taxa de chegadas, tempo de serviço, prioridade etc.

Por exemplo, um sistema de processamento de consultas que aceita apenas um tipo de consulta SQL; ou uma rede de comunicação que aceita apenas pacotes IP de serviço de "melhor esforço".

#### Múltiplas cargas

O sistema de filas pode aceitar diversos tipos de cargas ao mesmo tempo.

Por exemplo, o sistema de bancos de dados de uma empresa aceita consultas de cadastros, inserção de registros etc. Cada tipo de carga pode ter um tempo diferente para processamento, prioridade, taxa de chegadas de usuários etc.

Um exemplo de múltiplas cargas é o caso de uma rede de filas mista, onde uma carga é do tipo aberta e outra carga é do tipo fechada.

## 4.3 Variáveis Operacionais

As variáveis operacionais utilizadas nos estudos de redes de filas devem incluir os parâmetros de qual servidor elas se referem, já que uma rede de filas terá mais de um servidor.
São definidas as seguintes variáveis operacionais para os sistemas de redes de filas:

- $A_i$ número de chegadas de transações ao servidor *i*.
- $C_{ij}$ número de transações que saem do servidor *i* e vão para o servidor *j*.
- $C_{io}$ número de transações que saem do servidor *i* e deixam o sistema.
- $C_o$ número de transações que deixam o sistema.
- $V_i$ taxa relativa de visitas ao servidor *i*. Equivale ao número médio de visitas ao dispositivo por requisição.

$$V_i = \frac{C_i}{C_o}$$

- $D_i$ demanda de serviço no servidor *i*. Diferentemente do tempo de serviço, que representa o tempo de serviço para processamento de uma requisição, a demanda representa o tempo de serviço total para todas as visitas da mesma requisição ao servidor *i*.

$$D_i = V_i S_i$$

- $X$ ou $X_o$ taxa de processamento ou vazão do sistema.

$$X = \frac{C_o}{T}$$

Algumas variáveis operacionais estão representadas na Figura 4.5.

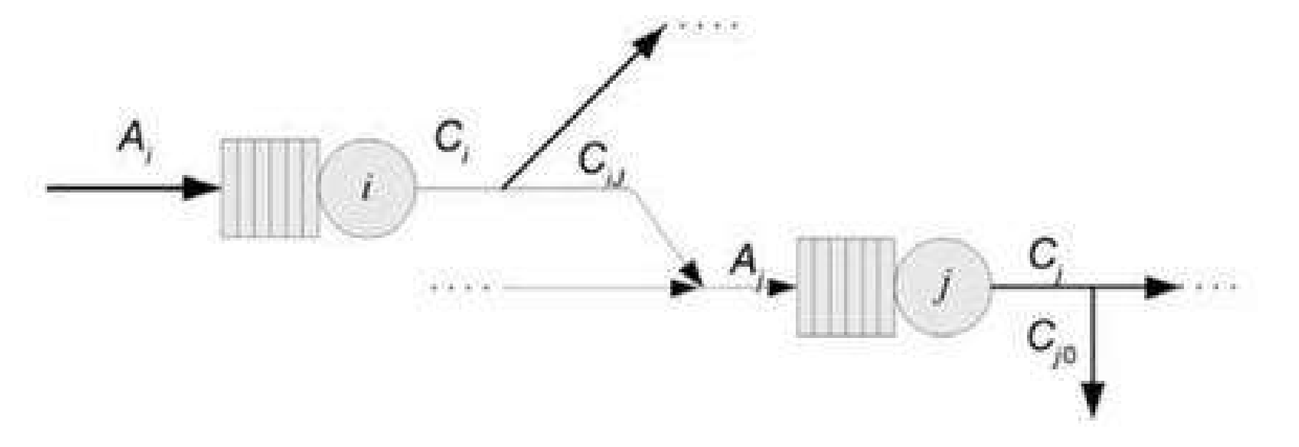


**Figura 4.5** Exemplo de uso das variáveis operacionais.

As variáveis operacionais vistas no Capítulo 3 sobre filas isoladas também são válidas em redes de filas. A representação de cada variável deve indicar qual centro de serviço ela caracteriza. Portanto, tem-se:

- $B_i$ tempo ocupado do servidor *i*.
- $S_i$ tempo de serviço do servidor *i*.
- $R_i$ tempo de resposta do servidor *i*.

## 4.4 Leis Operacionais para Redes de Filas

Nesta seção serão vistas três novas leis operacionais para uso dos sistemas de redes em filas: Lei do Fluxo Forçado, Lei do Tempo de Resposta e Lei da Demanda de Serviço.

### *4.4.1 Lei do Fluxo Forçado*

Esta lei mostra que o fluxo (vazão) em todas as partes do sistema deve ser proporcional.

$$X_i = \frac{C_i}{T} = \frac{C_i}{C_o} \times \frac{C_o}{T} = V_i X_o$$

**Exemplo 1**

Durante um período de 10 segundos, 40 requisições foram atendidas por um servidor de arquivos. Cada requisição requer dois acessos ao disco. O tempo médio de serviço no disco é de 30 milissegundos. Qual é a utilização média do disco nesse período?

Valores fornecidos no problema:

$T_{\text{observado}} = 10$ segundos
$C_i = 40$ requisições
$V_i = 2$
$S_i = 30$ ms $= 0{,}03$ segundo

Obtém-se

$$X_o = \frac{C_i}{T} = \frac{40}{10} = 4 \text{ requisições/segundo}$$

Como é desejado saber a utilização do disco, mas os dados fornecidos oferecem dados sobre a vazão do sistema pela Lei do Fluxo Forçado é possível obter a vazão do disco, e com ela, obter a utilização:

$$X_i = X_o V_i = 4 \times 2 = 8 \text{ req/s}$$
$$U_i = X_i S_i = 8 \times 0{,}03 = 0{,}24 = 24\%$$

A utilização do disco é de 24%.

**Exemplo 2**

Um sistema distribuído tem um servidor de impressão com velocidade de 60 páginas por segundo. Durante um período de tempo observado de 20 minutos, observou-se que o servidor imprimiu 500 páginas. Sabendo que toda requisição que chega ao sistema para impressão deve gerar cinco páginas impressas, qual foi o índice de tarefas completadas no sistema?

Valores fornecidos no problema:

$T_{observado} = 10$ minutos $= 20 \times 60 = 1200$ segundos
$C_i = 500$ páginas
$V_i = 5$

$$S_i = \frac{1}{\mu} = \frac{1}{60} = 0{,}017 \text{ segundo}$$

Obtém-se

$$X_i = \frac{C_i}{T} = \frac{500}{1200} = 0{,}42 \text{ impressão/segundo}$$

Como o problema oferece dados de um item do sistema (o servidor), mas espera-se obter os valores de vazão do sistema (composto de outros itens fora o servidor), pode-se usar a Lei do Fluxo Forçado para esse cálculo.

$$X_i = X_o V_i \Rightarrow X_o = \frac{X_i}{V_i} = \frac{0{,}42}{5} = 0{,}084 \text{ requisição/segundo}$$

## 4.4.2 Lei do Tempo de Resposta

O tempo de resposta total do sistema (o tempo de processamento de uma requisição, desde a entrada no sistema de filas até sua saída) depende do tempo de resposta de cada fila *versus* o número de visitas que a mesma requisição faz ao servidor daquela fila. Portanto, para um sistema com $K$ sistemas de fila:

$$R = \sum_{i=1}^{K} V_i \cdot R_i$$

Juntando a Lei do Tempo de Resposta com o Teorema do Tempo de Resposta (Capítulo 3), tem-se:

$$R = \sum_{i=1}^{K} \frac{D_i}{1 - U_i}$$

**Exemplo**

Considere o sistema de computação da Figura 4.6, que recebe transações a $\lambda = 5$ transações/segundo.

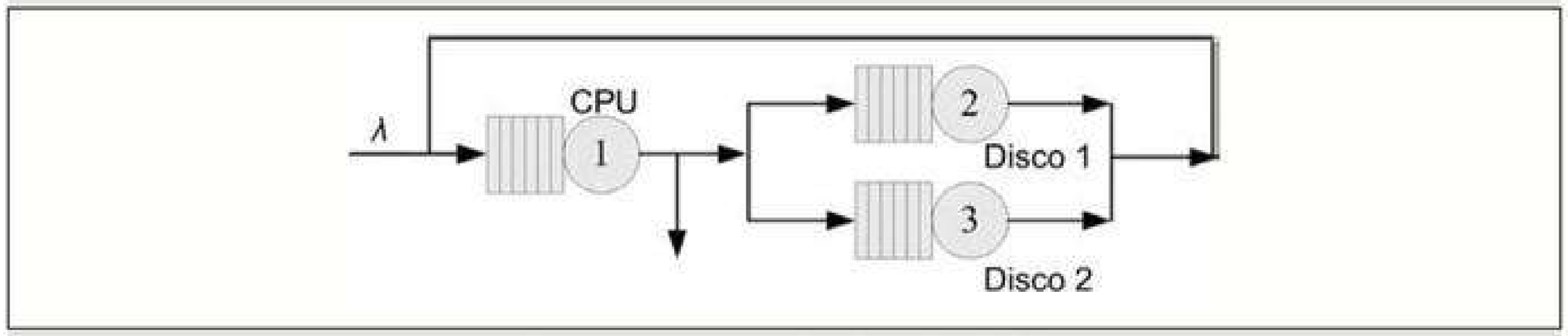


Figura 4.6 Exemplo de rede de fila.

Foram obtidos os seguintes dados:

$S_1 = 12$ ms/visita
$V_1 = 6$ visitas/transação
$S_2 = 30$ ms/visita
$V_2 = 1$ visita/transação
$S_3 = 25$ ms/visita
$V_3 = 4$ visitas/transação

Calcule o tempo médio de resposta do sistema.

Para calcular o tempo de resposta do sistema, é necessário obter o tempo de resposta de cada elemento do sistema, como a CPU e os dois discos. Portanto, sabendo apenas os valores de $\lambda$, $Vi$ e $Si$, pode-se calcular a demanda e utilização de cada elemento, e assim, calcular o tempo de resposta do sistema, através da segunda opção da Lei do Tempo de Resposta.

$D_1 = V_1 S_1 = 6 \times 0{,}012 = 0{,}072$ segundo
$D_2 = V_2 S_2 = 1 \times 0{,}03 = 0{,}03$ segundo
$D_3 = V_3 S_3 = 4 \times 0{,}025 = 0{,}1$ segundo

Para calcular a utilização de cada elemento, usa-se a Lei do Fluxo Forçado e assume-se que o tempo de serviço de cada disco, com múltiplas visitas, é sua demanda:

$U_1 = X \cdot D_1 = 5 \cdot 0{,}072 = 0{,}36$
$U_2 = X \cdot D_2 = 5 \cdot 0{,}03 = 0{,}15$
$U_3 = X \cdot D_3 = 5 \cdot 0{,}1 = 0{,}5$

$$R = R_1 + R_2 + R_3 = \frac{D_1}{1-U_1} + \frac{D_2}{1-U_2} + \frac{D_3}{1-U_3}$$

$$R = \frac{0{,}072}{1-0{,}36} + \frac{0{,}03}{1-0{,}15} + \frac{0{,}1}{0{,}5} = 0{,}11 + 0{,}035 + 0{,}2 = 0{,}345 \text{ s}$$

O tempo médio de resposta do sistema é de 0,345 segundo.

### 4.4.3 Lei da Demanda de Serviço

A demanda de serviço, anteriormente definida como $D_i = V_i D_i$, pode ser relacionada à vazão do sistema e sua utilização. Combinando as Leis do Fluxo Forçado e da Utilização, tem-se que:

$$D_i = V_i S_i = \frac{X_i}{X_o} \times \frac{U_i}{X_i} = \frac{U_i}{X_o}$$

**Exemplo 1**

Um sistema de computação foi monitorado durante 60 minutos. Nesse período, 120 transações foram processadas, e foram obtidos valores para a utilização da CPU e dos dispositivos de E/S. Calcule a demanda de serviço para cada dispositivo.

| **Dispositivo** | $U_i$ |
|---|---|
| CPU | 80% |
| Disco 1 | 20% |
| Disco 2 | 25% |
| Disco 3 | 30% |

Valores fornecidos no problema:

$T_{\text{observado}}$ = 60 minutos = 3600 segundos
$C_o$ = 120 transações

Obtém-se:

$$X_o = \frac{C_o}{T} = \frac{120}{3600} = 0{,}033 \text{ transação/segundo}$$

Para calcular a demanda de cada dispositivo, usa-se a Lei da Demanda de Serviço:

$$D_{\text{CPU}} = \frac{U_{\text{CPU}}}{X_o} = \frac{0{,}8}{0{,}033} = 24{,}24 \text{ s}$$

$$D_1 = \frac{U_1}{X_o} = \frac{0{,}2}{0{,}033} = 24{,}24 \text{ s}$$

$$D_2 = \frac{U_2}{X_o} = \frac{0{,}25}{0{,}033} = 7{,}58 \text{ s}$$

$$D_3 = \frac{U_3}{X_o} = \frac{0{,}3}{0{,}033} = 9{,}1 \text{ s}$$

**Exemplo 2**

Um sistema de computação possui dois tipos de carga, uma interativa e uma em lote. O sistema tem 20 terminais ativos e tempo de pensar de 15 segundos. A taxa de chegadas de requisições da carga interativa é de 2,5 transações por segundo. Em determinado disco *i* do sistema, a utilização total foi de 90% e o consumo de recursos foi de 0,4 segundo para a carga interativa e 0,2 segundo para a carga em lote. Qual o tempo médio de resposta da carga interativa?

Valores fornecidos no problema:

$N = 20$ terminais

$Z = 15$ segundos

$\lambda_{\text{interativo}} = 2{,}5$ transações/segundo

$U_i = 0{,}9$

$D_{i,\ \text{interativa}} = 0{,}4$ segundo

$D_{i,\ \text{lote}} = 0{,}2$ segundo

Para o cálculo do tempo de resposta da carga interativa, deve-se usar a Lei do Tempo de Resposta Interativo (Capítulo 3):

$$R = \frac{N}{X_o} - Z$$

Neste problema tem-se os valores de $N$ e $Z$, e precisa-se de $X_o$. Temos o valor da utilização do disco *i* e os valores de demanda de cada carga nesse dispositivo.

Portanto:

$$D_i = \frac{U_i}{X_o}$$

Como $U_{i,\ \text{interativa}} + U_{i,\ \text{lote}} = U_i$

$D_{i,\ \text{interativa}} X_{o,\ \text{interativa}} + D_{i,\ \text{lote}} X_{o,\ \text{lote}} = U_i$

Pela Lei do Fluxo Forçado, tem-se

$0{,}4 \times X_{o,\ \text{interativa}} + 0{,}2 \times 2{,}5 = 0{,}9$

$X_{o,\ \text{interativa}} = 1$ transação/segundo

Portanto:

$$R = \frac{20}{1} - 15 = 5\,\text{s}$$

O tempo de resposta do sistema é de 5 segundos.

## 4.5 Exercícios

1) Transações de um banco de dados executam uma média de 4,5 operações de E/S no servidor do BD. O servidor foi monitorado durante uma hora e, nesse período, 7200 transações foram executadas. Qual a vazão média do disco do servidor? Se cada operação de E/S no disco leva 20 milissegundos em média, qual a utilização do disco?

2) Um servidor de banco de dados tem uma CPU e dois discos e recebe requisições a uma taxa de 10.800 requisições por hora. Cada requisição necessita de 20 milissegundos de CPU e realiza cinco operações de E/S no Disco 1 e três operações no Disco 2. Cada operação de E/S demora em média 15 milissegundos. Qual o tempo médio de resposta por requisição, vazão média do servidor e utilização da CPU e discos?

3) Para o sistema *timesharing* da Figura 4.7, o tempo de pensar do usuário foi observado em cinco segundos, os tempos de serviço de tarefas dos discos foram 30 e 25 milissegundos, e o tempo médio de serviço da tarefa na CPU de 40 milissegundos.

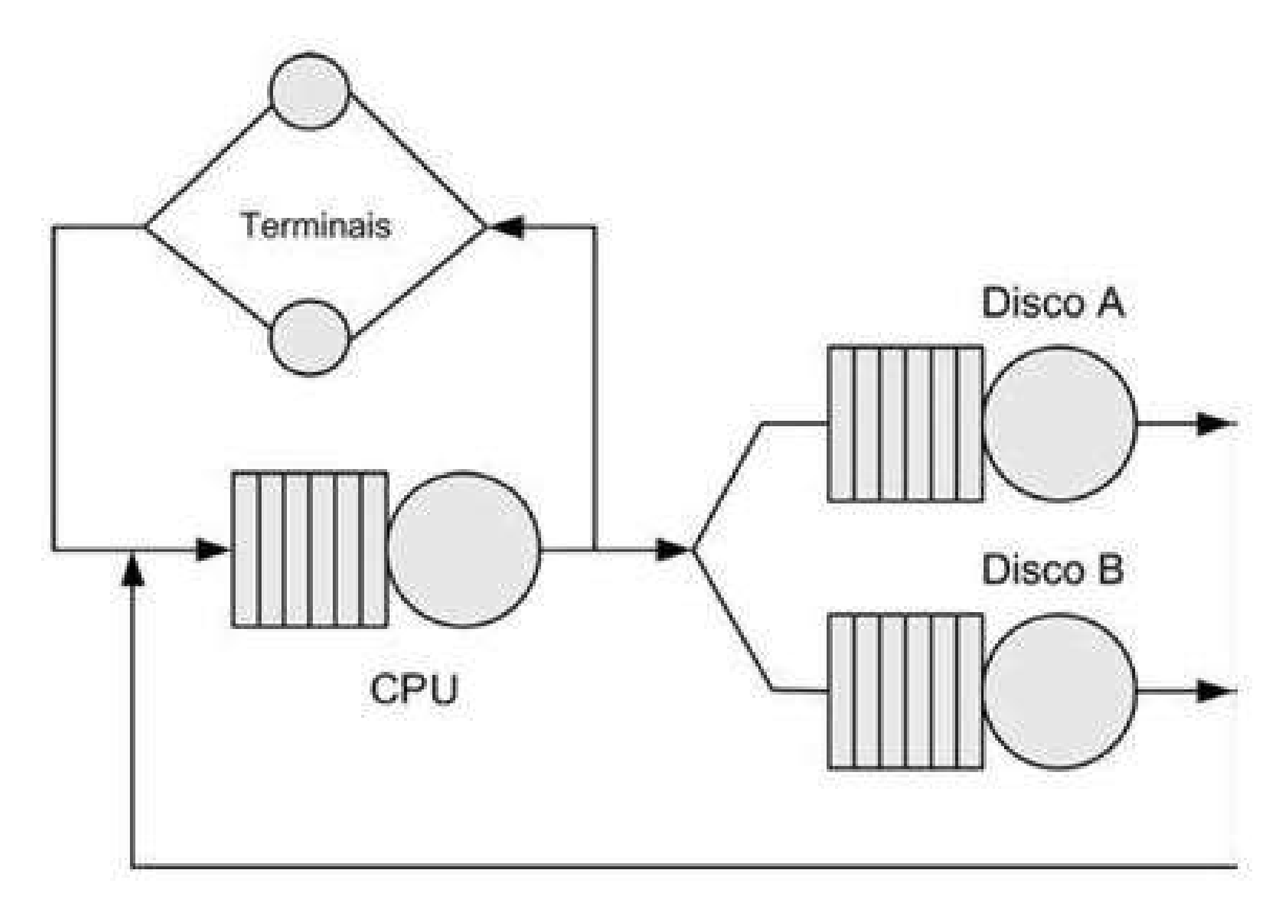


**Figura 4.7** Sistema *timesharing*.

Responda:

a) Para cada dispositivo, qual foi a demanda de serviços total?
b) Se a utilização do Disco A é de 60%, qual a utilização da CPU e do disco?
c) Se a utilização do Disco B é de 10%, qual o tempo médio de resposta quando se tem 20 usuários no sistema?
d) Qual o tempo médio de resposta do sistema?

4) Sobre o sistema do Exercício 1, qual dispositivo seria o gargalo se:

   a) A CPU fosse trocada por outra duas vezes mais rápida?
   b) O Disco A fosse trocado por outro duas vezes mais lento?
   c) O Disco B fosse trocado por outro duas vezes mais lento?

5) Considere o sistema da Figura 4.8 com duas cargas de trabalho, *a* e *b*.

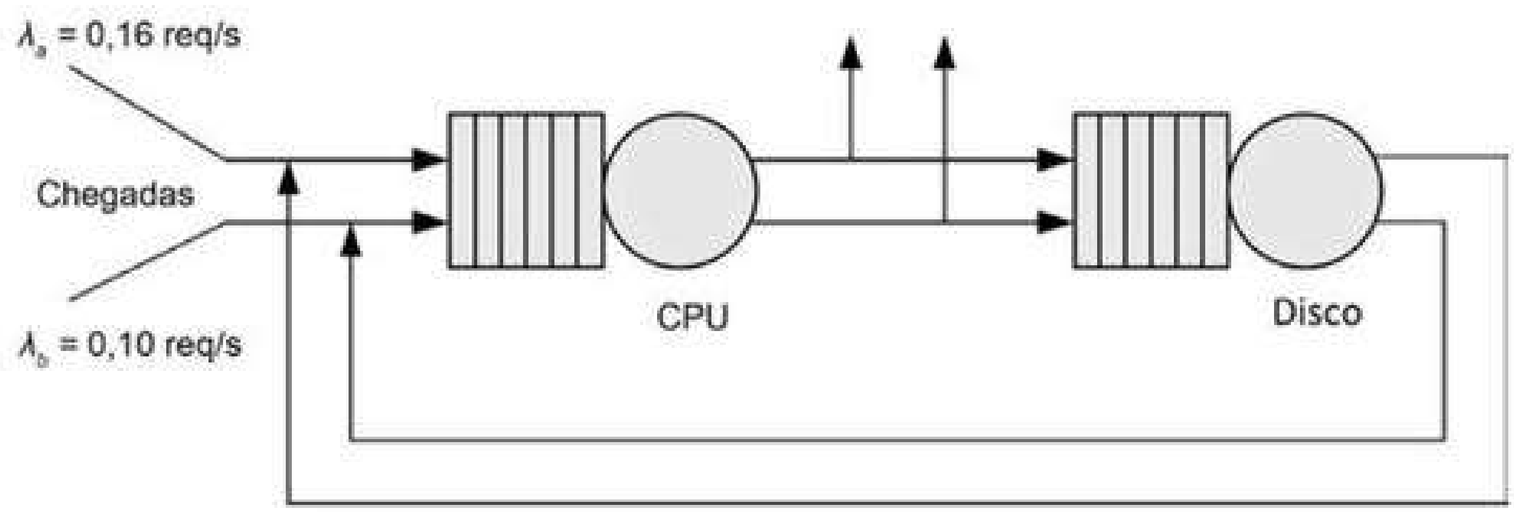


**Figura 4.8** Sistema com duas cargas.

Calcule a vazão, a utilização e o tempo de resposta da Carga *a* no sistema, sabendo que:

$V_{a,\text{CPU}} = 10$
$V_{a,\text{disco}} = 9$
$D_{a,\text{CPU}} = 1$ s
$D_{a,\text{disco}} = 3$ s
$V_{b,\text{CPU}} = 5$
$V_{b,\text{disco}} = 4$
$D_{b,\text{CPU}} = 2$ s
$D_{b,\text{disco}} = 4$ s

6) Considere o sistema computacional da Figura 4.6. Esse sistema processa duas cargas de trabalho: consulta a banco de dados e atualização de banco de dados. Os seguintes dados foram obtidos durante 60 minutos de observação:

| | Carga 1 (consulta) | Carga 2 (atualização) |
|---|---|---|
| Taxa de chegadas | 6 | 2 |
| $U_{\text{CPU}}$ | 0,2 | 0,4 |
| $U_{\text{disco1}}$ | 0,4 | 0,3 |
| $U_{\text{disco2}}$ | 0,1 | 0,2 |

Calcule o tempo médio de resposta do sistema.

# Análise Quantitativa de Redes de Filas

# 5

## 5.1 Introdução

A análise quantitativa de filas isoladas, ou de redes de filas, descreve o comportamento dos modelos de acordo com sua evolução no tempo. Este capítulo é um complemento do material estudado na Análise Operacional de Redes de Filas, vista nos Capítulos 3 e 4.

Neste texto será visto como analisar quantitativamente os modelos de fila de uma ou de múltiplas classes de carga de trabalho, abertas ou fechadas.

## 5.2 Análise de Modelos com Carga Única

Os modelos de filas com um único tipo de carga de trabalho podem possuir cargas abertas ou fechadas.

### *5.2.1 Classes Abertas*

Como visto no Capítulo 4, as cargas abertas são as cargas do tipo transação. As requisições chegam do exterior, são processadas, e no final saem do sistema. O número de clientes no sistema é ilimitado.

Uma classe aberta é definida por dois parâmetros:

- $\lambda$ taxa de chegadas de clientes no sistema,
- $S_k$ tempo de serviço em cada servidor $k$.

A Figura 5.1 mostra um modelo de fila aberta, com apenas uma classe de clientes chegando ao sistema a uma taxa $\lambda$, que passará por $K$ centros de serviço, cujos tempos de serviço são $S_1$, $S_2$, ..., $S_k$.

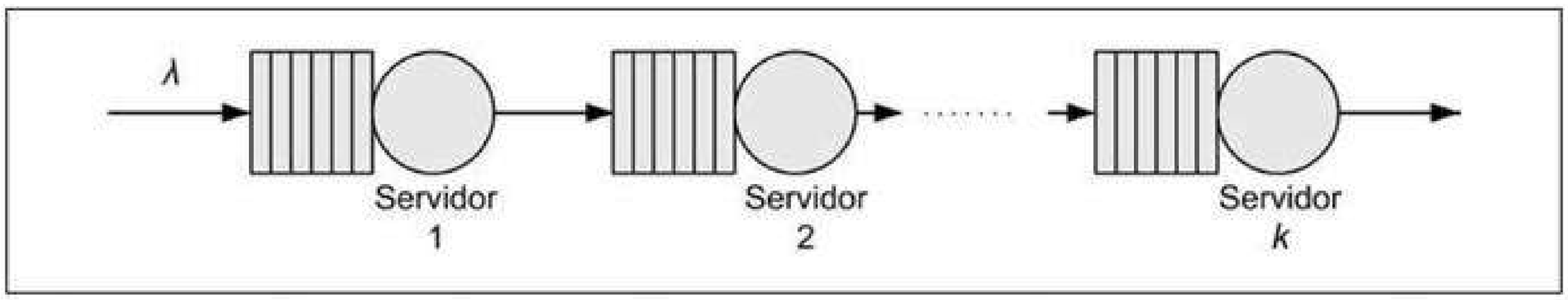


**Figura 5.1** Exemplo de rede de filas com somente um tipo de carga e *K* servidores.

As principais características de classes abertas são:

## Capacidade de processamento

λ é o número médio de transações que chegam por unidade de tempo e é independente do número de transações no sistema. A capacidade de processamento é a taxa de chegadas que satura o sistema. Essa taxa é dada por:

$$\lambda_{sat} = \frac{1}{D_{máx}}$$

onde a demanda máxima do sistema $(D_{máx})$ é a maior demanda de um dado servidor *k* do sistema. Admite-se que $\lambda < \lambda_{sat}$.

## Vazão

A vazão em um centro de serviço *k* depende do número de visitas de cada requisição a esse servidor. De acordo com a Hipótese do Equilíbrio de Fluxo (Capítulo 4) tem-se:

$$X_k(\lambda) = \lambda V_k$$

## Utilização

Pela Lei da Utilização (Capítulo 4), a utilização em um servidor *k* é igual à vazão multiplicada pelo tempo de serviço:

$$U_k(\lambda) = X_k(\lambda) S_k = \lambda D_k$$

## Tempo de resposta

O tempo de resposta, ou tempo de residência, no servidor k é o tempo total gasto pelo cliente, entrando na fila aguardando seu atendimento e o tempo de processamento. Portanto:[1]

$$R_k(\lambda) = \begin{cases} D_k & \text{centros de atraso} \\ \dfrac{D_k}{1 - U_k(\lambda)} & \text{centros de processamento} \end{cases}$$

[1] As deduções das fórmulas podem ser vistas em Lazowska(1984).

O tempo de resposta total do sistema pode ser calculado pela soma dos tempos de residência em todos os $K$ servidores:

$$R(\lambda) = \sum_{k=1}^{K} R_k(\lambda)$$

## Número de clientes

O número de clientes na fila em cada centro pode ser calculado pelo uso da Lei de Little:

$$(\lambda) = \lambda R_k(\lambda) = \begin{cases} U_k & \text{centros de atraso} \\ \dfrac{U_k(\lambda)}{1 - U_k(\lambda)} & \text{centros de processamento} \end{cases}$$

O número total de clientes no sistema pode ser calculado pela Lei de Little ou pela soma da fila total em todos os $K$ servidores:

$$Q(\lambda) = \lambda R(\lambda) = \sum_{k=1}^{K} Q_k(\lambda)$$

**Exemplo**

Na rede de filas da Figura 5.2, requisições obedecem a um processo de chegadas com média de cinco requisições por segundo.

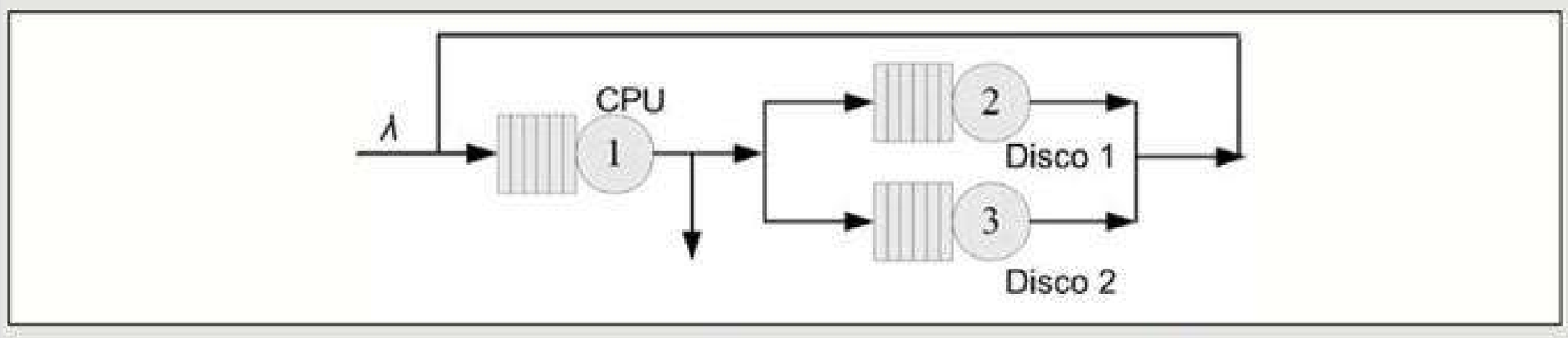


Figura 5.2 Rede de fila com três centros e uma classe.

Na monitoração do sistema foram obtidos os seguintes dados para $S_k$ e $V_k$:

$S_1 = 6$ milissegundos
$S_2 = 20$ milissegundos
$S_3 = 30$ milissegundos
$V_1 = 5$ visitas
$V_2 = 2$ visitas
$V_3 = 2$ visitas

*(continua)*

Calculando a demanda dos servidores, tem-se:

$$D_1 = V_1 S_1 = 5 \times 0{,}006 = 0{,}03 \text{ segundo}$$
$$D_2 = V_2 S_2 = 2 \times 0{,}02 = 0{,}04 \text{ segundo}$$
$$D_3 = V_3 S_3 = 2 \times 0{,}03 = 0{,}06 \text{ segundo}$$

Calculando a capacidade de processamento:

$$\lambda_{sat} = \frac{1}{D_{máx}} = \frac{1}{0{,}06} = 16{,}7 \text{ requisições/segundo}$$

Com esse valor, verificamos que $\lambda < \lambda_{sat}$.

- Vazão dos servidores:

$$X_1(\lambda) = \lambda V_1 = 5 \times 5 = 25 \text{ requisições/s}$$
$$X_2(\lambda) = \lambda V_2 = 5 \times 2 = 10 \text{ requisições/s}$$
$$X_3(\lambda) = \lambda V_3 = 5 \times 2 = 10 \text{ requisições/s}$$

- Utilização dos servidores:

$$U_1(\lambda) = X_1(\lambda) D_1 = 25 \times 0{,}03 = 0{,}75 = 75\%$$
$$U_2(\lambda) = X_2(\lambda) D_2 = 10 \times 0{,}04 = 0{,}4 = 40\%$$
$$U_3(\lambda) = X_3(\lambda) D_3 = 10 \times 0{,}06 = 0{,}6 = 60\%$$

- Tempo de resposta dos servidores:

$$R_1(\lambda) = D_1 - U_1(\lambda) = 0{,}031 - 0{,}75 = 0{,}12 \text{ segundo}$$

$$R_2(\lambda) = D_2 - U_1(\lambda) = 0{,}041 - 0{,}4 = 0{,}067 \text{ segundo}$$

$$R_3(\lambda) = D_2 - U_1(\lambda) = 0{,}061 - 0{,}6 = 0{,}15 \text{ segundo}$$

- Tempo de resposta do sistema:

$$R(\lambda) = \sum_{k=1}^{3} R_k(\lambda) = R_1\lambda + R_2\lambda + R_3\lambda = 0{,}12 + 0{,}067 + 0{,}15 = 0{,}337 \text{ segundo}$$

- Número de requisições de clientes na fila de cada servidor:

$$Q_1(\lambda) = \lambda R_1(\lambda) = 5 \times 0{,}12 = 0{,}6 \text{ requisição}$$

$$Q_2(\lambda) = \lambda R_2(\lambda) = 5 \times 0{,}067 = 0{,}335 \text{ requisição}$$

$$Q_3(\lambda) = \lambda R_3(\lambda) = 5 \times 0{,}15 = 0{,}75 \text{ requisição}$$

- Número de clientes na fila no sistema:

$$Q(\lambda) = \lambda R(\lambda) = \sum_{k=1}^{3} Q_k(\lambda) = Q_1(\lambda) + Q_2(\lambda) + Q_3(\lambda) = 0{,}6 + 0{,}335 + 0{,}75 = 1{,}685 \text{ requisição}$$

### 5.2.2 Classes Fechadas

Como visto no Capítulo 4, as cargas fechadas são as cargas do tipo transação. O número de requisições dos clientes no sistema é fixo e elas trafegam no sistema, são processadas, e retornam até o usuário. O número de clientes no sistema é limitado.

A carga de trabalho, caracterizada como fechada, possui as características de:

- Intensidade da carga é dada pela população de clientes, $N$, que é o número de requisições concorrentes em execução. Este número de clientes no sistema é limitado e conhecido.
- Clientes possuem tempo de pensar $Z$.
- Tempo de serviço em cada centro $k$ é dado por $S_k$.

O sistema da Figura 5.3 mostra um sistema fechado com uma carga, composto de três ($K = 3$) servidores, com tempos de serviço $S_1$, $S_2$ e $S_3$.

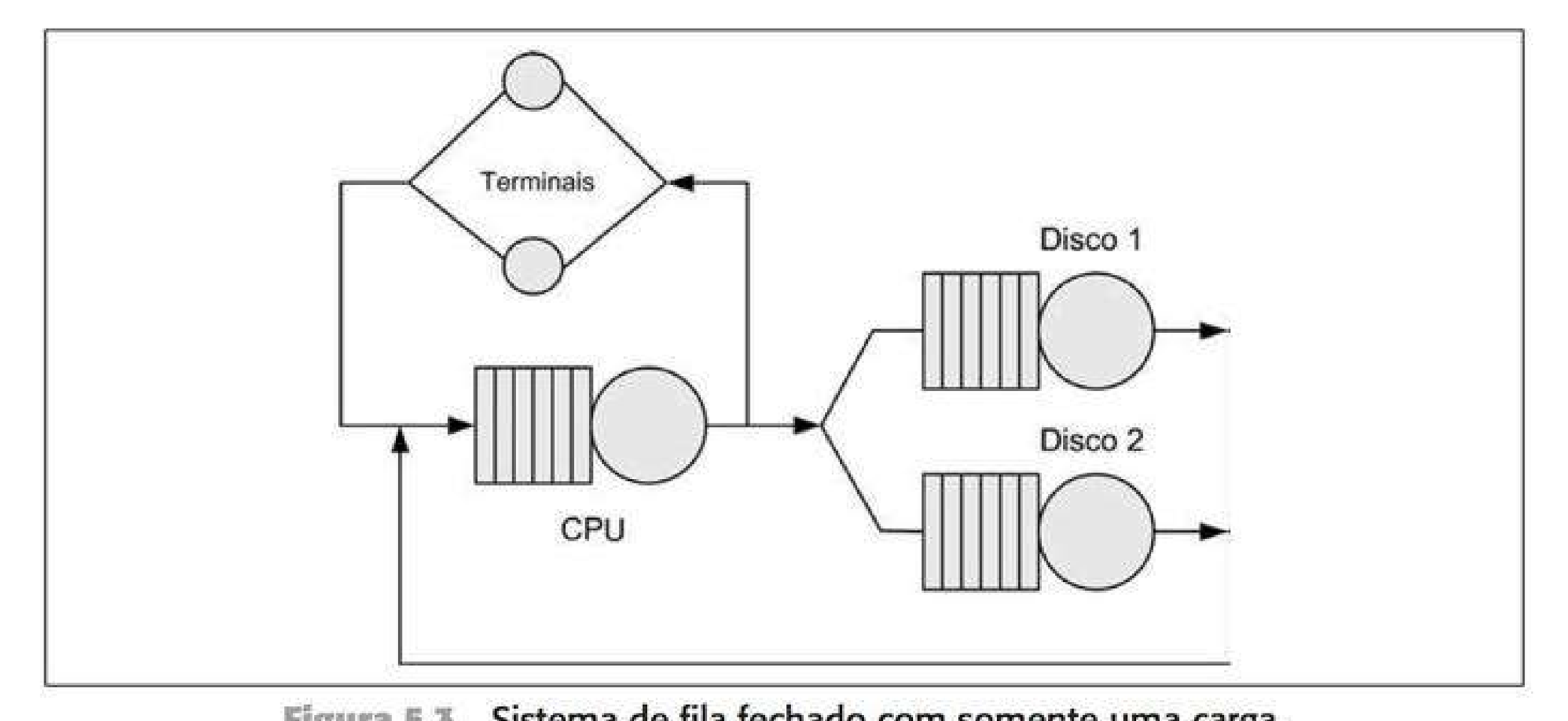


**Figura 5.3** Sistema de fila fechado com somente uma carga.

As redes de classe fechada devem ser analisadas pelo algoritmo de Análise do Valor Médio (AVM).

#### Algoritmo de análise do valor médio

Nesta seção será apresentado o algoritmo de AVM de cálculo para uma carga.

As seguintes definições são usadas para cálculo da AVM:

- $\boldsymbol{N}$ número de transações no sistema de filas fechado.
- $\boldsymbol{R_k(N)}$ tempo médio de resposta por visita no servidor $k$:

$$R_k(N) = \begin{cases} D_k & \text{Centros de atraso} \\ D_k[1+Q_k(N)] & \text{Centros de processamento} \end{cases}$$

onde $Q_k(N)$ é o número médio de clientes vistos no servidor $k$ quando clientes chegam ao servidor em questão.

- $X(N)$ taxa de processamento ou vazão do sistema:

$$X(N) = \frac{N}{Z + \sum_{k=1}^{K} R_k(N)}$$

- $Q_k(N)$ comprimento da fila no servidor $k$:

$$Q_k(N) = X(N) R_k(N)$$

- $R_o(N)$ tempo médio de resposta total do sistema.

Todas as variáveis relacionadas assumem que existem $N$ requisições em todo o sistema.

O algoritmo de AVM para apenas um tipo de carga obtém os valores de $R_k(N)$, $X(N)$ e $Q_k(N)$ para qualquer valor de $N$ no sistema. Usa-se a demanda de serviços em cada servidor nos cálculos e o número de filas no sistema $(K)$.

O algoritmo é iterativo, começando com $N = 1$ e $Q_k(0) = 0$, para cada instante de tempo.

| | | |
|---|---|---|
| **Passo 1** | Faça $Q_i(0) = 0$ para $i = 1...K$ | *Inicialização* |
| **Passo 2** | Para $n = 1, ..., N$ calcule para $i = 1, ..., K$ | *Iteração* |
| **(a)** | $R_k(n) = \begin{cases} D_k & \text{centros de atraso} \\ D_k[1+Q_k(N)] & \text{centros de processamento} \end{cases}$ | |
| **(b)** | $X(n) = \dfrac{n}{Z + \sum_{k=1}^{K} R_k(n)}$ | |
| **(c)** | $Q_k(n) = X(n) R_k(n)$ | |
| **Passo 3** | Pare | |

**Exemplo**

No modelo de sistema representado na Figura 5.3, cada pedido do usuário efetua nove acessos à CPU ($V_{CPU} = 9$), três acessos ao Disco A ($V_A = 3$) e cinco acessos ao disco B ($V_B = 5$). Os tempos de serviço por visita aos Discos A e B são, respectivamente, 200 e 250 milissegundos e o tempo de serviço da CPU é de 25 milissegundos. A demanda em cada dispositivo é $D_{CPU} = 0{,}225$ segundo, $D_A = 0{,}6$ segundo e $D_B = 1{,}25$ segundo.

*(continua)*

Utilizando o AVM para cálculo das medidas de desempenho, tem-se:

**Inicialização:**

Número de usuários: $N = 0$

Comprimento das filas dos dispositivos: $Q_{CPU} = 0$, $Q_A = 0$, $Q_B = 0$

**Iteração 1:**

Número de usuários: $N = 1$

Tempos de resposta dos dispositivos:

$$R_{CPU} = D_{CPU}(1 + Q_{CPU}) = 0{,}225(1 + 0) = 0{,}225 \text{ s}$$

$$R_A = D_A(1 + Q_A) = 0{,}6\ (1 + 0) = 0{,}6 \text{ s}$$

$$R_B = D_B(1 + Q_B) = 1{,}25\ (1 + 0) = 1{,}25 \text{ s}$$

Tempo de resposta do sistema:

$$R_o = R_{CPU} + R_A + R_B = 0{,}225 + 0{,}6 + 1{,}25 = 2{,}075 \text{ s}$$

Vazão do sistema:

$$X = \frac{N}{R_o + Z} = \frac{1}{2{,}075 + 4} = 0{,}16 \text{ pedido/segundo}$$

Comprimento das filas dos dispositivos:

$$Q_{CPU} = XR_{CPU} = 0{,}16 \times 0{,}225 = 0{,}036 \text{ pedido}$$

$$Q_A = XR_A = 0{,}16 \times 0{,}6 = 0{,}096 \text{ pedido}$$

$$Q_B = XR_B = 0{,}16 \times 1{,}25 = 0{,}2 \text{ pedido}$$

**Iteração 2:**

Número de usuários: $N = 2$

Tempos de resposta dos dispositivos:

$$R_{CPU} = D_{CPU}\left(1 + Q_{CPU}\right) = 0{,}225\ (1 + 0{,}036) = 0{,}23$$

$$R_A = D_A\left(1 + Q_A\right) = 0{,}6\ (1 + 0{,}096) = 0{,}66 \text{ s}$$

$$R_B = D_B\left(1 + Q_B\right) = 1{,}25\ (1 + 0{,}2) = 1{,}5 \text{ s}$$

Tempo de resposta do sistema:

$$R = R_{CPU} + R_A + R_B = 0{,}23 + 0{,}66 + 1{,}5 = 2{,}39 \text{ s}$$

Vazão do sistema:

$$X = \frac{N}{R_o + Z} = \frac{2}{2{,}39 + 4} = 0{,}31 \text{ pedido/segundo}$$

*(continua)*

Comprimento das filas dos dispositivos:

$$Q_{CPU} = XR_{CPU} = 0{,}31 \times 0{,}23 = 0{,}07 \text{ pedido}$$

$$Q_A = XR_A = 0{,}31 \times 0{,}66 = 0{,}2 \text{ pedido}$$

$$Q_B = XR_B = 0{,}31 \times 1{,}5 = 0{,}46 \text{ pedido}$$

E assim por diante, até o valor de $N$ desejado.

## 5.3 Análise de Redes com Múltiplas Cargas

Diversos modelos de filas utilizam mais de uma carga de trabalho para poderem ter mais controle sobre os parâmetros de entrada e obtenção de dados de desempenho mais precisos para cada tipo de carga. Por exemplo, através das técnicas de QoS (Qualidade de Serviços), uma rede de computadores pode dar prioridade para tráfego de clientes *Premium* e menor prioridade para os clientes que não pagam pela preferência. Num sistema de filas para essa rede, o tempo de serviço em cada nó para os clientes *Premium* seria menor do que o tempo de serviço para clientes comuns.

Os modelos com múltiplas cargas também devem ser analisados para cargas abertas ou fechadas.

### 5.3.1 Classes Abertas

A Figura 5.4 mostra um modelo aberto de múltiplas classes, contendo $C$ classes ($a$ e $b$, com $C = 2$) e $K$ servidores. A classe $a$ trafega apenas pelo lado superior do modelo, enquanto a classe $b$ trafega apenas pelo lado inferior.

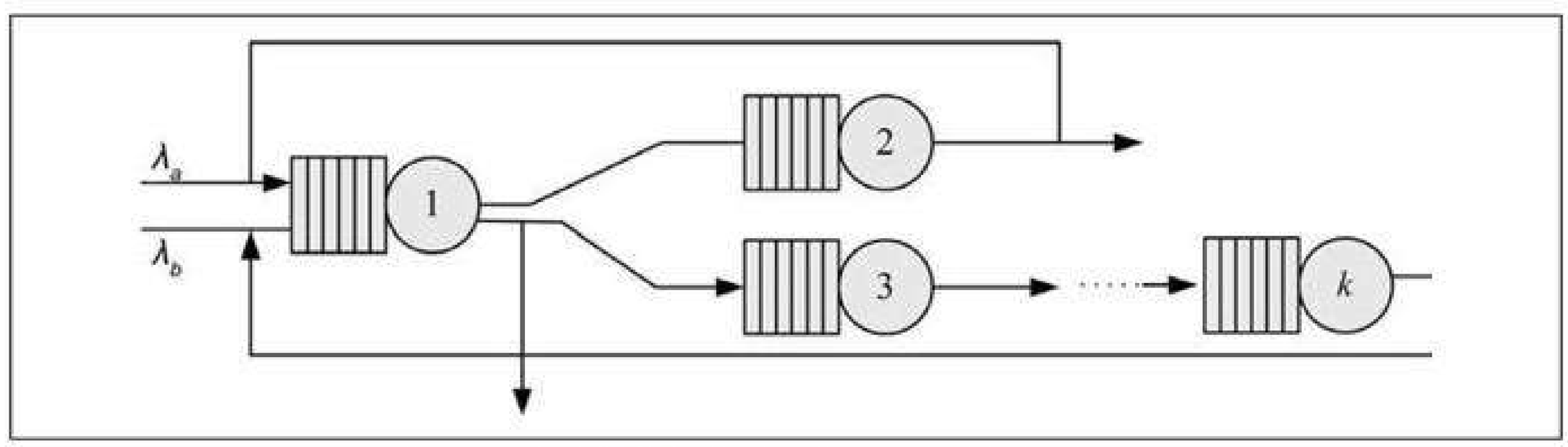


**Figura 5.4** Modelo aberto com $C = 2$ classes.

As múltiplas classes de carga aberta possuem diferentes taxas de chegadas para cada classe $c$. Usa-se o vetor $\lambda$ para denotar o vetor das taxas de chegadas das $C$ classes:

$$\vec{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_C)$$

As principais características de classes abertas são:

### Capacidade de processamento

Existe suficiente capacidade em um servidor para atender a todas as classes se:

$$\text{máx}_k \left\{ \sum_{c=1}^{C} \lambda_c D_{c,k} \right\} < 1$$

Nenhum servidor deve ser saturado pela carga de todas as classes, portanto deve-se satisfazer a inequação.

### Vazão

A vazão de uma classe *c* em um centro de serviço *k* é uma função de $\vec{\lambda}$

$$X_{c,k}\left(\vec{\lambda}\right) = \lambda_c V_{c,k}$$

### Utilização

Pela Lei da Utilização, tem-se:

$$U_{c,k}\left(\vec{\lambda}\right) = X_{c,k}\left(\vec{\lambda}\right) S_{c,k} = \lambda_c D_{c,k}$$

### Tempo de resposta

O tempo de resposta, ou tempo de residência no servidor *k*, é dado por:

$$R_{c,k}\left(\vec{\lambda}\right) = \begin{cases} D_{c,k} & \text{centros de atraso} \\ \dfrac{D_{c,k}}{1 - \sum_{i=1}^{C} U_{i,k}\ \vec{\lambda}} & \text{centros de processamento} \end{cases}$$

O tempo de resposta total do sistema pode ser calculado pela soma dos tempos de residência em todos os *K* servidores:

$$R_c\left(\vec{\lambda}\right) = \sum_{k=1}^{K} R_{c,k}(\vec{\lambda})$$

### Número de clientes

O número de clientes na fila, em cada centro, pode ser calculado pelo uso da Lei de Little nas equações de tempo de residência anteriores:

$$Q_{c,k}\left(\vec{\lambda}\right) = \lambda_c R_{c,k}\left(\vec{\lambda}\right) = \begin{cases} U_{c,k}(\vec{\lambda}) & \text{Centros de atraso} \\ \dfrac{U_{c,k}(\vec{\lambda})}{1 - \sum_{i=1}^{C} U_{i,k}(\vec{\lambda})} & \text{Centros de processamento} \end{cases}$$

O número total de clientes no sistema pode ser calculado pela Lei de Little ou pela soma da fila total para todas as classes $C$ em todos os $K$ servidores:

$$Q_c\left(\vec{\lambda}\right) = \lambda_c R_c\left(\vec{\lambda}\right) = \sum_{k=1}^{K} Q_{c,k}(\vec{\lambda})$$

**Exemplo**

Um sistema computacional analisado possuía duas cargas de trabalho, A e B. Desejando calcular o tempo de resposta do sistema para as diferentes cargas, o gerente do sistema mensurou a demanda de recursos em cada dispositivo e obteve os seguintes dados:

| Dispositivo | Carga A | | Carga B | |
|---|---|---|---|---|
| | $D_A$ | $V_A$ | $D_B$ | $V_B$ |
| CPU | 1,2 | 2 | 0,7 | 1 |
| Disco 1 | 1,3 | 2 | 0,8 | 3 |

O vetor de taxa de chegadas medido foi $\vec{\lambda} \equiv (0{,}3\,;\,0{,}5)$ requisições por segundo

Calculando a vazão, tem-se:

$$X_{A,\text{CPU}}\left(\vec{\lambda}\right) = \lambda_A V_{A,\text{CPU}} = 0{,}3 \times 2 = 0{,}6 \text{ requisição/segundo}$$

$$X_{A,D1}\left(\vec{\lambda}\right) = \lambda_A V_{A,D1} = 0{,}3 \times 2 = 0{,}6 \text{ requisição/segundo}$$

$$X_{A,\text{CPU}}\left(\vec{\lambda}\right) = \lambda_B V_{B,\text{CPU}} = 0{,}5 \times 1 = 0{,}5 \text{ requisição/segundo}$$

$$X_{A,D1}\left(\vec{\lambda}\right) = \lambda_B V_{B,D1} = 0{,}5 \times 3 = 1{,}5 \text{ requisição/segundo}$$

Pela Lei da Utilização, tem-se:

$$U_{A,\text{CPU}}\left(\vec{\lambda}\right) = \lambda_A D_{A,\text{CPU}} = 0{,}3 \times 1{,}2 = 0{,}3 = 36\%$$

$$U_{A,D1}\left(\vec{\lambda}\right) = \lambda_A D_{A,D1} = 0{,}3 \times 1{,}3 = 0{,}39 = 39\%$$

$$U_{B,\text{CPU}}\left(\vec{\lambda}\right) = \lambda_B D_{B,\text{CPU}} = 0{,}5 \times 0{,}7 = 0{,}35 = 35\%$$

$$U_{B,D1}\left(\vec{\lambda}\right) = \lambda_B D_{B,D1} = 0{,}5 \times 0{,}8 = 0{,}4 = 40\%$$

*(continua)*

Calculando o tempo de resposta para cada carga, em cada servidor, tem-se:

$$R_{A,\text{CPU}}(\vec{\lambda}) = \frac{D_{A,\text{CPU}}}{1-\sum_{i=1}^{C} U_{i,\text{CPU}}(\vec{\lambda})} = \frac{D_{A,\text{CPU}}}{1-(U_{A,\text{CPU}}(\vec{\lambda})+U_{B,\text{CPU}}(\vec{\lambda}))} = \frac{1{,}2}{1-(0{,}36+0{,}35)} = 4{,}14 \text{ segundos}$$

$$R_{A,D1}(\vec{\lambda}) = \frac{D_{A,D1}}{1-(U_{A,D_1}(\vec{\lambda})+U_{B,D_1}(\vec{\lambda}))} = \frac{1{,}3}{1-(0{,}39+0{,}40)} = 6{,}19 \text{ segundos}$$

$$R_{B,\text{CPU}}(\vec{\lambda}) = \frac{D_{B,D1}}{1-(U_{A,\text{CPU}}(\vec{\lambda})+U_{B,\text{CPU}}(\vec{\lambda}))} = \frac{0{,}7}{1-(0{,}36+0{,}35)} = 2{,}41 \text{ segundos}$$

$$R_{B,D1}(\vec{\lambda}) = \frac{D_{\text{B},D1}}{1-(U_{A,D_1}(\vec{\lambda})+U_{B,D_1}(\vec{\lambda}))} = \frac{0{,}8}{1-(0{,}39+0{,}40)} = 3{,}81 \text{ segundos}$$

Calculando o tempo de resposta do sistema para cada carga, tem-se:

$$R_A(\vec{\lambda}) = R_{A,\text{CPU}}(\vec{\lambda}) + R_{A,D1}(\vec{\lambda}) = 4{,}14 + 6{,}19 = 10{,}33 \text{ segundos}$$

$$R_B(\vec{\lambda}) = R_{B,\text{CPU}}(\vec{\lambda}) + R_{B,D1}(\vec{\lambda}) = 2{,}41 + 3{,}81 = 6{,}22 \text{ segundos}$$

Portanto, a carga B possui tempo de resposta menor, isto é, é mais rápida que a carga A.

### *5.3.2 Classes Fechadas*

Para o cálculo das métricas de desempenho dos modelos com classes de carga fechada, usa-se o algoritmo de Análise de Valor Médio Aproximado. Esse algoritmo pode ser estudado em Lazowska (1984) e Menascé (1985).

## 5.4 Exercícios

1) Um servidor Web possui quatro discos com os dados a serem consultados. O sistema foi observado por uma hora e nesse período 120 transações foram processadas. Um monitor de software capturou informações sobre a utilização dos dispositivos do sistema e as mostrou em uma tabela:

| Dispositivo | $U_k$ |
|---|---|
| Disco 1 | 0,25 |
| Disco 2 | 0,40 |
| Disco 3 | 0,36 |

| | |
|---|---|
| Disco 4 | 0,05 |

Identifique qual dispositivo é o gargalo do sistema e calcule o tempo de resposta do sistema.

2) Para o sistema *timesharing* da Figura 5.3, o tempo de pensar do usuário foi observado em cinco segundos, os tempos de serviço de tarefas dos discos foram de 30 e 25 milissegundos, e o tempo médio de serviço da tarefa na CPU foi de 40 milissegundos. Calcule o número médio de requisições em cada servidor, utilizando o algoritmo de AVM.

3) O sistema do Exercício 1 foi modificado para atender a três cargas de trabalho. Os seguintes dados foram obtidos em 60 minutos de observação:

| Dispositivo | Carga A | Carga B | Carga C |
|---|---|---|---|
| Taxa de chegadas | 6 | 2 | 4 |
| $U_{Disco1}$ | 0,20 | 0,40 | 0,20 |
| $U_{Disco2}$ | 0,40 | 0,12 | 0,20 |
| $U_{Disco3}$ | 0,22 | 0,05 | 0,20 |
| $U_{Disco4}$ | 0,05 | 0,18 | 0,20 |

Calcule o tempo de resposta das três cargas em cada dispositivo.

# Parte III
# Simulação


6 Modelagem via Simulação 89

7 Simulação Aplicada a Redes de Computadores 93

8 Executando e Analisando a Simulação 101

# Modelagem via Simulação

# 6

Analisar o desempenho de um sistema normalmente é uma tarefa ligada ao planejamento de alguma grande ação e, quase sempre, precede uma decisão envolvendo vultosa soma de investimentos. Tais investimentos podem ser quantificados tanto financeiramente, como no caso de empresas que pretendam expandir seu parque operacional, quanto intelectualmente, no caso de instituições acadêmicas onde muitos recursos são aplicados em pesquisa. Sob esse prisma, analisar o desempenho de um sistema se torna uma tarefa de grande responsabilidade, uma vez que seus resultados podem influenciar sobremaneira as estratégias gerenciais da instituição.

A análise de desempenho de um sistema computacional pode ser feita basicamente de duas formas: por aferição, também referenciada como experimentação, ou por modelagem (Law, 2000). Como mencionado no Capítulo 1, alguns autores desmembram em três formas, considerando a simulação como uma técnica em separado.

A abordagem por aferição consiste basicamente na instalação de mecanismos de medição no sistema existente ou em seu protótipo e será discutida no Capítulo 7. Se o sistema não existir ou não estiver disponível, então é necessário que seja modelado. Por vezes, mesmo o sistema estando disponível, a modelagem se torna a única alternativa para a realização de testes de desempenho em larga escala. Por exemplo, é comum propostas inovadoras de protocolos de rede serem apresentadas na academia para uso na Internet. Todavia, por ser uma rede em produção, com milhares de usuários conectados, não há como realizar testes sem inserir um risco desnecessário à estabilidade do sistema. Nesse caso, modelar pode vir a ser a melhor opção.

O processo de modelagem ainda pode ser classificado em modelagem física e modelagem matemática. Um avião construído em escala menor e submetido a um túnel de vento para estudo do efeito do movimento do ar é um exemplo de modelagem física. Já um modelo matemático ainda pode ser estudado a partir de uma solução analítica ou por simulação. Um modelo analítico bastante simplificado que poderia ser usado em diversos segmentos da engenharia é $P = F/A$ (Pressão = Intensidade da Força/Área da Superfície). A partir de duas dessas variáveis (Força e Área, por exemplo), pode-se chegar à pressão necessária e, com isso, delinear o desempenho de um determinado sistema. Obviamente, existem situações bastante complexas. Porém, quando o sistema se torna complexo demais para ser modelado analiticamente, a solução pode estar no uso de simulação. Nesse caso, as linguagens de programação constituem uma ferramenta de grande utilidade pela flexibilidade e pelos recursos nativos que nelas são disponibilizados.

Por via de regra, modelar um sistema computacional requer o uso de uma representação simplificada. Por isso, se faz necessário um alto grau de abstração, desprezando-se os detalhes de menor relevância. Isso, contudo, não deve comprometer a macroestrutura do sistema em estudo. Com a modelagem via simulação, por exemplo, diversos parâmetros podem ser mudados, testados e analisados repetidas vezes, o que pode vir a diminuir substancialmente os custos do projeto. Outra grande contribuição de uma boa modelagem é a facilidade de ajustes no sistema. Modelar um sistema não significa reproduzi-lo em cada detalhe. Caso contrário, não o estaríamos modelando e sim construindo o próprio sistema. Se, por exemplo, codificarmos os mínimos detalhes de funcionamento de um roteador, estaremos na verdade construindo esse roteador e não modelando.

Saber o limiar que precisa ser imposto no grau de abstração é um dos fatores que caracteriza um bom projetista de simulação.

Na opção por simulação ainda se deve ter em mente que os modelos podem ser dinâmicos ou estáticos, determinísticos ou estocásticos e contínuos ou discretos.

Os sistemas dinâmicos envolvem variações contínuas no tempo e normalmente são caracterizados pela retroalimentação, ou seja, os resultados produzidos na saída são usados como *feedback* para ajustes na entrada. O mecanismo de envio de pacotes do protocolo TCP é um exemplo de um sistema dinâmico, uma vez que ajusta a taxa de envio em função das notificações de reconhecimento (ACKs) que recebe. Por outro lado, os sistemas estáticos atuam em um instante de tempo com valores previamente especificados.

Quando o modelo não contém nenhuma probabilidade, ou seja, não há o caráter randômico, ele é classificado como determinístico. Usualmente os sistemas que se enquadram nessa classificação são fechados, ou seja, o sistema é isolado da influência de variáveis externas. Já se essa influência ocorrer ele é dito probabilístico ou estocástico.

No caso de uma simulação em sistemas discretos, ocorrem eventos associados a instantes de tempo com variação limitada a um conjunto de pontos finitos e predefinidos, o que já não ocorre em um sistema contínuo onde as variáveis de estado mudam continuamente no tempo. Na prática, poucos sistemas são totalmente discretos ou contínuos. A forma de melhor caracterizá-los é que varia.

## 6.1 Motivação para Simular

A principal motivação para se usar simulação é a praticidade. Essa técnica permite que tanto os cenários, quanto os parâmetros que o caracterizam sejam alterados com pouco nível de complexidade. Se a ideia é analisar o desempenho de uma proposta diante de diferentes situações (mudando tráfego, protocolos, escalabilidade etc.) tem-se na simulação uma grande aliada. Normalmente os processos de simulação computacional demandam grandes recursos computacionais. Todavia, cada vez mais isso deixa de ser um grande problema. Segundo Banks (2004), um indicativo de que o processo de simulação não é apropriado é quando o problema pode ser resolvido por senso comum. Nesse caso, a solução analítica pode ser bem mais indicada.

## 6.2 Conceitos

De maneira genérica, simulação pode ser considerada como o ato de reproduzir algum processo por meio de um recurso análogo, especialmente com a finalidade de estudo. Os processos de simulação podem ser divididos em discretos e contínuos. Como os simuladores computacionais trabalham com sinais discretos, dado que os sinais contínuos são formados por infinitos pontos, a ênfase deste livro será dada em simulações discretas. Isso pode ser feito com a ajuda de um programa de computador.

Quando se pensa em criar, e não somente em usar um simulador, cabe citar uma recomendação de Shannon (1989) que sugere que 40% do tempo a ser investido na concepção de um simulador devem ser consumidos na definição, projeto e modelagem do problema, 20% devem ser reservados na concepção dos programas cujos elementos conceituais foram definidos na primeira fase e os 40% restantes devem ser reservados para verificar e validar o modelo simulado. Para se construir um simulador, alguns elementos e técnicas básicas precisam ser previamente compreendidos.

### *6.2.1 Simulações Dependentes de Tempo*

A grande maioria dos simuladores utiliza um controle cronológico de eventos para a execução das atividades. Essa concepção pressupõe a existência de contador de tempo, assim como um limiar dentro do qual tudo acontece. Nesse caso, deve-se informar esse limiar como um dos parâmetros da simulação ao final do qual todos os processos serão encerrados. O controle da simulação é feito por um gerente chamado de *kernel*, ou núcleo do simulador, que aciona os blocos de simulação associados à técnica utilizada.

As simulações dependentes de tempo, ou *time-dependent simulations*, normalmente podem ser aplicadas de várias formas, mas aqui serão destacadas duas. A primeira é conhecida como simulação dirigida a tempo, ou *time-driven simulation*, e a segunda como simulação dirigida a eventos, ou *event-driven simulation*. Ambas baseadas em variações de tempo, mas com comportamentos ligeiramente diferentes.

#### *Time-driven simulation*

Na simulação dirigida a tempo (*time-driven simulation*) a linha de tempo até o limiar especificado é dividida em intervalos iguais. Existe, nesse caso, uma variável que grava o instante de tempo atual e o incrementa em intervalos fixos. Após cada incremento são verificadas as pendências e executados os eventos que nele ocorreram. Supondo que esse intervalo ($\Delta t$) seja de meio segundo e o limiar de simulação seja de três segundos, teríamos uma escala com {0}, {0.5}, {1.0}, {1.5}, {2.0}, {2.5}, {3.0}. Em cada um desses intervalos o escalonador entra em ação para executar todos os eventos pendentes. Se, por exemplo, dois eventos ocorrerem nos instantes {0.7} e {0.8}, respectivamente, eles serão executados no instante {1.0} da simulação. Se vários eventos forem agendados para o mesmo tempo de execução, então a disciplina da fila se encarregará de organizar a prioridade. Se, por exemplo, a disciplina for FIFO (*First In First Out*) então eles serão organizados de forma que o primeiro a chegar será o primeiro a ser executado.

Para melhor exemplificar será utilizada uma analogia com um parque de diversões. Se um dos brinquedos desse parque fosse dirigido a tempo (*time-driven*) seria definido um intervalo

fixo ($\Delta t$ = 10 minutos, p. ex.) dentro do qual a operadora (escalonador) iria permitir que todos entrassem, após o que acionaria o brinquedo novamente. Os que chegassem em seguida teriam que aguardar em fila até que os 10 minutos seguintes transcorressem. Passado esse tempo, a operadora voltaria novamente à fila para atender os que ali aguardassem, e assim por diante.

### *Event-driven simulation*

Em contra partida, na simulação dirigida a eventos (*event-driven simulation*) não há um intervalo de tempo predefinido. Se, por exemplo, dois eventos ocorrerem nos instantes {0.7} e {0.8}, respectivamente, eles serão executados nesses mesmos instantes da simulação. No exemplo, anterior se o atendimento do parque fosse dirigido a eventos (*event-driven*), os usuários seriam conduzidos ao brinquedo sob demanda, conforme fossem chegando. No Quadro 6.1 são apresentados os algoritmos básicos das duas técnicas aqui discutidas.

**QUADRO 6.1** Algoritmos básicos das técnicas de simulação

| ***Time-driven simulation*** | ***Event-driven simulation*** |
|---|---|
| Inicializar o estado do sistema e o tempo de simulação<br>Enquanto (simulação não terminada)<br>1. Coletar estatísticas sobre o estado corrente<br>2. Executar os eventos que ocorreram entre o último intervalo e o atual<br>3. Incrementar o tempo de simulação | Inicializar o estado do sistema<br>Inicializar a lista de eventos<br>Enquanto (simulação não terminada)<br>1. Coletar estatísticas sobre o estado corrente<br>2. Remover o primeiro evento da lista, executá-lo<br>3. Ajustar o tempo para o tempo desse evento |

# Simulação Aplicada a Redes de Computadores

# 7

Para consolidar os conceitos de simulação abordados neste livro, os exemplos serão aplicados à área de redes de computadores. O primeiro passo é identificar os elementos de um simulador de redes.

## 7.1 Elementos de um Simulador de Redes

De acordo com Issariyakul (2009), a estrutura de um simulador consiste em sete elementos básicos: entidades, recursos, eventos, escalonador, variáveis globais, gerador de números aleatórios e coletor de estatísticas.

As entidades são objetos que interagem entre si em um programa de simulação causando mudança no estado do sistema. Em uma simulação de redes a representação dos roteadores é um exemplo de entidade.

Já os recursos se caracterizam por apresentarem limitações como, por exemplo, a quantidade de largura de banda disponível em um enlace.

Os eventos podem ser caracterizados como acontecimentos planejados e coordenados com antecedência. O processo de enfileiramento de pacotes diante de uma disciplina de fila como FIFO (*First In First Out*) ou SFQ (*Stochastic Fairness Queuing*) caracteriza bem esse item, uma vez que a disciplina é previamente definida.

O escalonador faz um controle cronológico dos eventos que ocorrerão na simulação. É no escalonador, por exemplo, que se programa a queda de um enlace ou os instantes de início e término de uma transmissão.

Com as variáveis globais é possível acessar funções e modificar parâmetros em diversos pontos da simulação, permitindo, por exemplo, que a taxa de transmissão de um fluxo seja alterada em determinado instante.

O gerador de números aleatórios introduz randomicidade no processo de simulação, oferecendo resultados estatísticos mais confiáveis.

Finalmente, com o coletor de estatística os dados estatísticos relevantes são coletados e exibidos de forma amigável, após o processo de simulação ter sido concluído.

Os conceitos aqui apresentados serão mais bem explorados adiante, tomando como exemplo um dos maiores simuladores gratuitos já desenvolvidos e disponibilizados, o *Network Simulator*.

## 7.2 Entendendo o *Network Simulator* (NS-2)

O *Network Simulator* (NS) é um dos simuladores de rede mais referenciados no mundo. Dependente de tempo e dirigido a eventos (*event-driven*), foi concebido em 1989 a partir de uma variante de outro simulador, o *Real Network Simulator*. Em 1995, o desenvolvimento do NS foi incorporado à DARPA (Agência de Projetos de Pesquisa Avançados dos Estados Unidos) através do projeto VINT (*Virtual InterNetwork Testbed*). Posteriormente, com a nova versão, o simulador ficou conhecido como NS-2. O fato de ser gratuito e de código-fonte aberto tem permitido que milhares de pesquisadores incorporem e testem suas ideias no simulador, motivo pelo qual seu uso vem crescendo significativamente no meio acadêmico.

O simulador NS-2 é baseado em duas linguagens de programação: C++ e TCL (*Tool Command Language*). Isso se deve ao fato de nele existir dois segmentos bem definidos. Um atuando como *back-end* e outro como *front-end*. O primeiro (*back-end*) comporta a infraestrutura básica do simulador e, por conseguinte, requereu o uso de uma linguagem de programação mais robusta, principalmente porque o código não muda com frequência (C++ foi a opção escolhida pelos projetistas). Nesse segmento são codificados os protocolos, agentes, aplicações, dispositivos etc. Um exemplo são as variantes do protocolo TCP (Tahoe, Reno, New Reno etc.), que dificilmente são modificados. Havendo real necessidade de fazê-lo, devem ser realizados os ajustes e compilar novamente o simulador. Por outro lado, em um ambiente de simulação é comum o usuário realizar diversos testes em seus cenários, alterando enlaces, taxas de transmissão, protocolos de transporte, políticas de fila, enfim, a intermitência desses ajustes sugere o uso de um ambiente interpretado, pois os elementos do cenário de simulação serão modificados com bastante frequência e ressubmetidos, tornando o uso de uma linguagem compilada pouco produtivo. Assim se justifica a escolha da linguagem interpretada em TCL atuando como *front-end*.

O NS-2 oferece um alto nível de abstração para representar alguns elementos de rede. Exemplo disso é o *buffer* que, apesar de ser um dispositivo físico associado ao *hardware* (p. ex., roteador), encontra-se modelado como um componente do enlace (Figura 7.1).

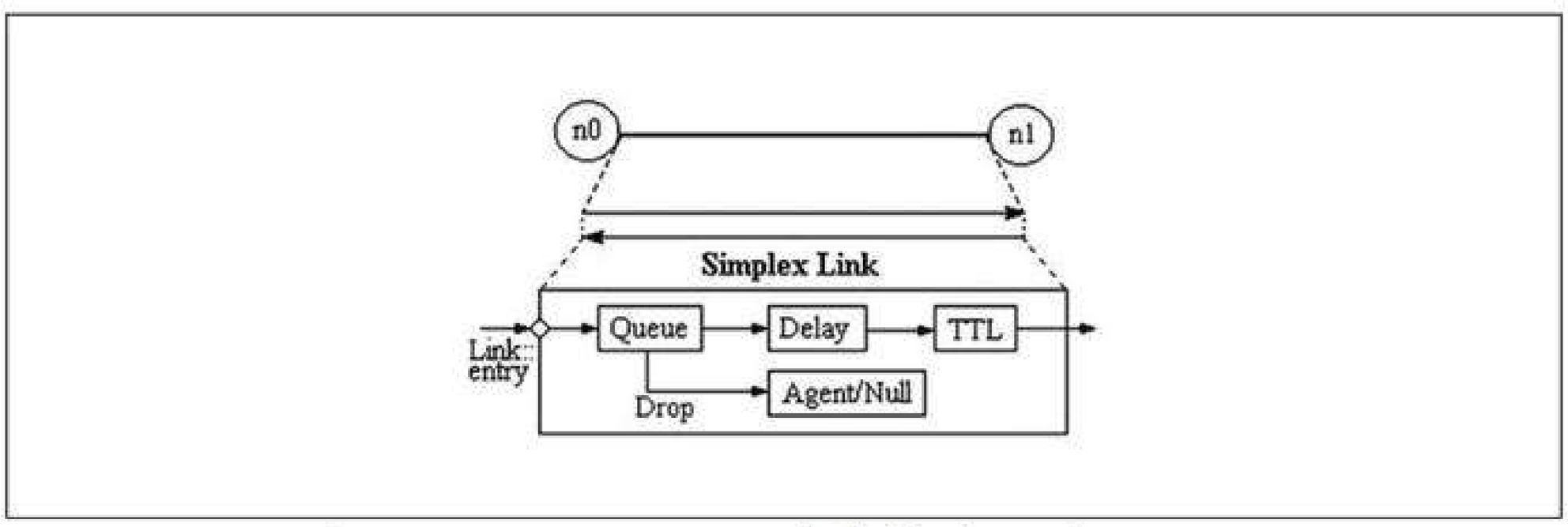


**Figura 7.1** Representação do *buffer* (*queue*) no NS-2.

## 7.2.1 Instalação e Configuração

O passo inicial é realizar o *download* do arquivo de instalação disponível no *source forge* (use como pista NS-2). Existem basicamente duas formas de fazê-lo: em partes (*in-peaces*), recomendada apenas para usuários mais experientes onde apenas os componentes necessários ao tipo de simulação que se pretende serão instalados, ou completo (*all-in-one*), onde o simulador é instalado com todos os seus módulos. O Apêndice A traz um guia para se instalar o NS-2 em distribuições derivadas do Debian.

## 7.2.2 Planejamento da Simulação

Planejar uma simulação é tarefa extremamente importante. Nem por isso, recursos mais sofisticados são requeridos. Papel, caneta e algumas ideias bem delineadas já resolvem. Para facilitar o entendimento serão propostos e planejados cenários simples de uma rede de grande arrasto (*Wide Area Network*). Didaticamente, fica mais fácil segmentar esse planejamento em etapas. Serão usadas cinco delas nesse exemplo: definição dos nós, configuração dos enlaces, especificação do tráfego, escalonamento dos fluxos e previsão de adversidades. Também é altamente recomendável esboçar graficamente o que se pretende, utilizando convenções que o ajudarão no processo de documentação e organização das ideias. Adiante serão apresentadas as etapas e o trecho de código em TCL utilizado para caracterizá-las no NS-2.

### Definição dos nós

Constitui a identificação individual dos nós (p. ex., *host* e roteadores) envolvidos na simulação, assim como seus atributos. No cenário exemplo (Figura 7.2) serão considerados roteadores localizados nas cidades do Rio de Janeiro, São Paulo, Brasília e Belém.

**Figura 7.2** Definição dos nós que compõem a simulação.

No NS-2, o instanciamento dos nós que representam os roteadores do cenário utilizado é feito conforme mostrado na Figura 7.3.

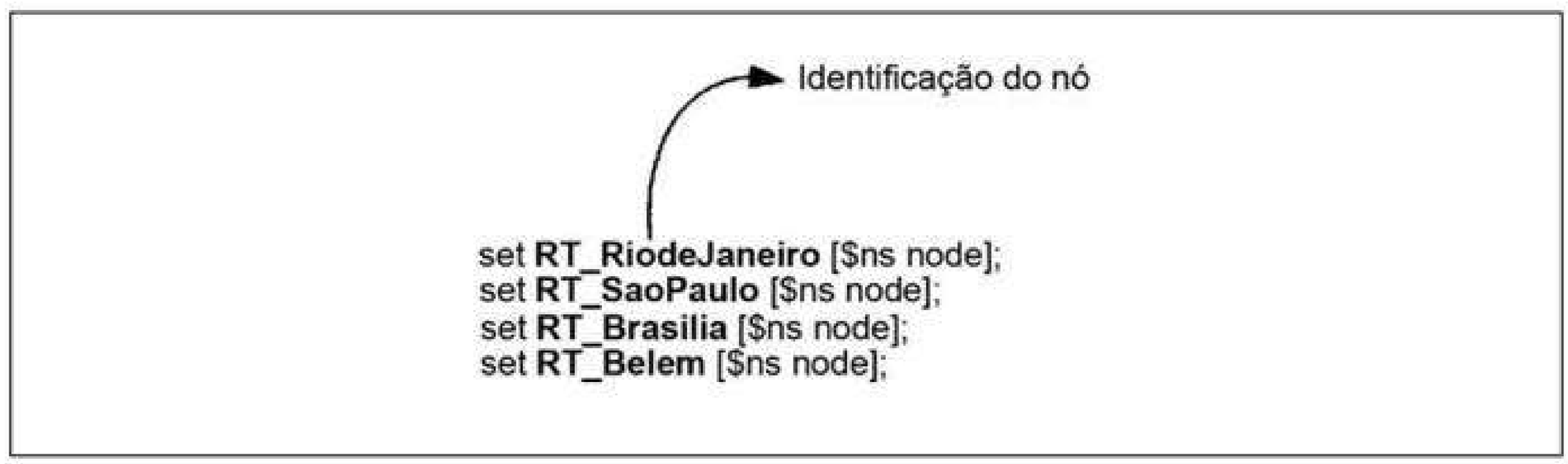


**Figura 7.3** Codificação dos nós no NS-2.

## Configuração dos enlaces

No NS-2, em vez de se informar uma mídia física específica, como cabo UTP ou fibra ótica, deve-se caracterizá-la (lembre-se de que simular implica trabalhar com abstrações). A combinação de alguns parâmetros permitirá definir, mesmo que implicitamente, o meio físico que se quer utilizar. Os parâmetros mais comumente usados em simulações de redes guiadas são a capacidade máxima de vazão suportada pelo canal ou largura de banda (*bandwidth*), o atraso ponto a ponto (*delay*) e o tipo do enlace (*simplex/duplex*), conforme ilustrado na Figura 7.4.

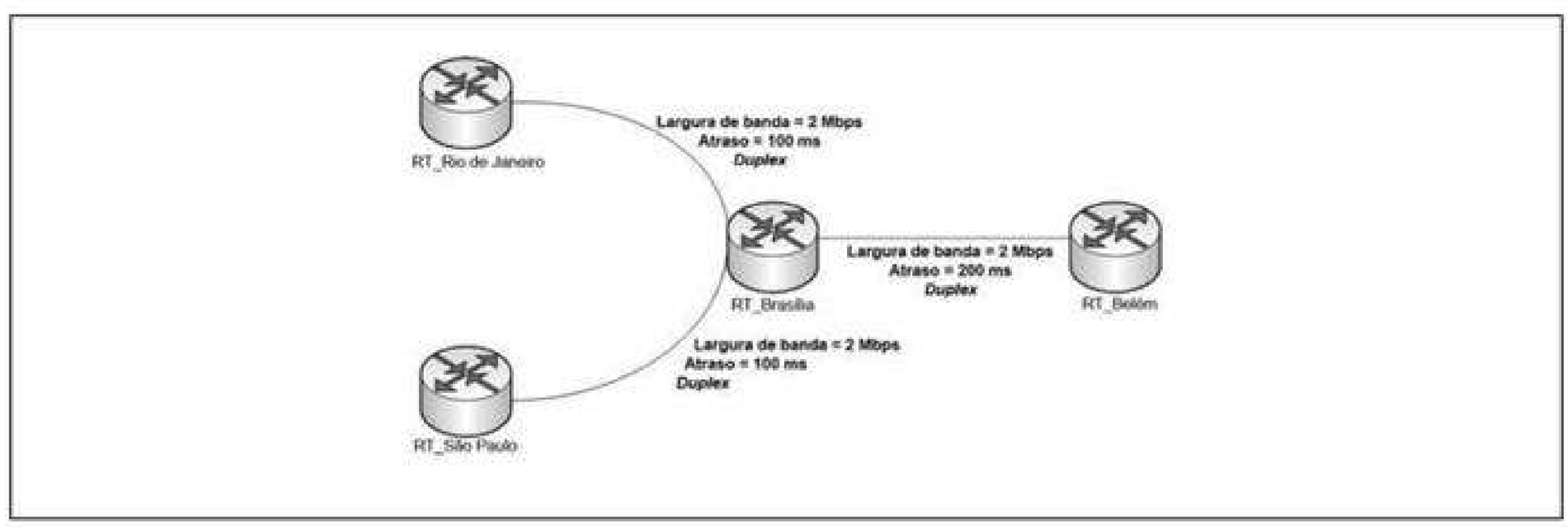


**Figura 7.4** Definição de enlaces que compõem a simulação.

Para configurar os enlaces, o NS-2 utiliza o padrão apresentado na Figura 7.5 dentro da seguinte ordem:

- Tipo de enlace (*simplex-link/duplex-link*). Lembrando que um canal *simplex* transmite dados em uma só direção. Já os canais *duplex* (*full*) transmitem em ambas as direções, simultaneamente.
- Nós, de onde e para onde o enlace parte, indicando os percursos ou rotas possíveis.

- Capacidade máxima de vazão do enlace. Normalmente expressa em Megabits por segundo (Mbps).
- Atraso inerente ao meio físico entre o ponto de saída e o ponto de chegada do enlace. Normalmente expresso em milissegundos (ms).
- Política de fila (p.ex., DropTail, SFQ, RED etc.), que indica a disciplina de encaminhamento de pacotes que será adotada no *buffer* dos roteadores. Durante as situações de congestionamento isso será um grande diferencial no comportamento da rede.

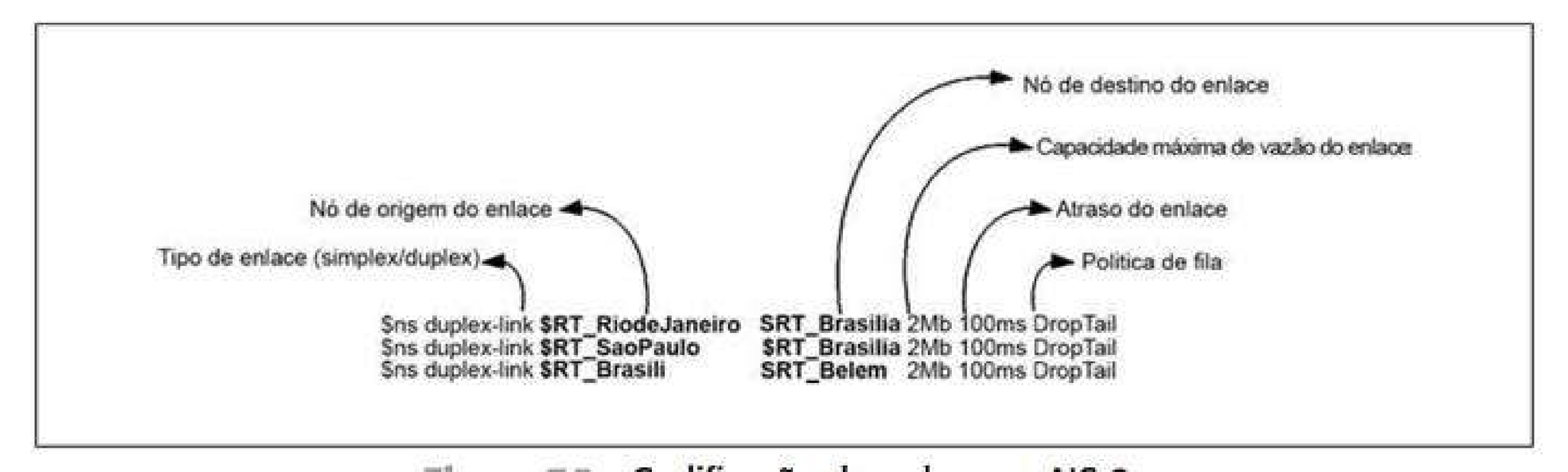


**Figura 7.5** Codificação do enlace no NS-2.

## Especificação do tráfego

Criado o leiaute da rede, ou seja, a topologia a ser adotada, deve-se decidir sobre o tráfego. Nessa fase, dentre outros, precisa-se indicar quem manda, o que se manda, para quem se manda e de que forma, além de especificar protocolos, aplicações e parâmetros. No contexto do cenário apresentado na Figura 7.6 serão realizadas duas transmissões de aplicações CBR,[1] uma agendada para o início da simulação (instante 1) e outra para o meio da simulação (instante 15). Ambas terão como destino o roteador de Belém (RT_Belém), uma será transmitida do Rio de Janeiro e outra de São Paulo. No NS-2 se faz necessária a criação de um agente responsável por receber os dados no destino. No caso de a aplicação vir a ser transportada pelo protocolo UDP, o agente de recepção se chamará agente nulo (new Agent/Null). O nome nulo é dado porque não há retroalimentação para o sistema transmissor. Se o transporte, por outro lado, for feito pelo protocolo TCP e, portanto, precisar gerar um pacote de reconhecimento (ACK), o agente de recepção se chamará *Sink* (new Agent/TCPSink). Um evento *Sink*, também referenciado como ouvinte (*listener*), normalmente é um trecho de código que define como o *host* vai obter os dados de um evento.

Em uma simulação de redes é sempre recomendável que se criem pontos de congestionamento (gargalos) no cenário. Sem eles, não haverá problemas na rede e, por conseguinte, pouco haverá para se analisar. A Figura 7.6 mostra uma situação típica onde, em determinado

[1] Aplicações CBR (*Constant Bit Rate*) são usadas para representar uma transmissão com taxa constante. Exemplo disso seria a *stream* de vídeo não compactado em tempo real.

momento da simulação, as transmissões de fluxos entrantes no roteador de Brasília (4 Mbps), provenientes dos roteadores de São Paulo e do Rio de Janeiro, excedem a capacidade máxima do enlace de saída (2 Mbps), gerando um ponto de gargalo e, consequentemente, promovendo descarte de pacotes.

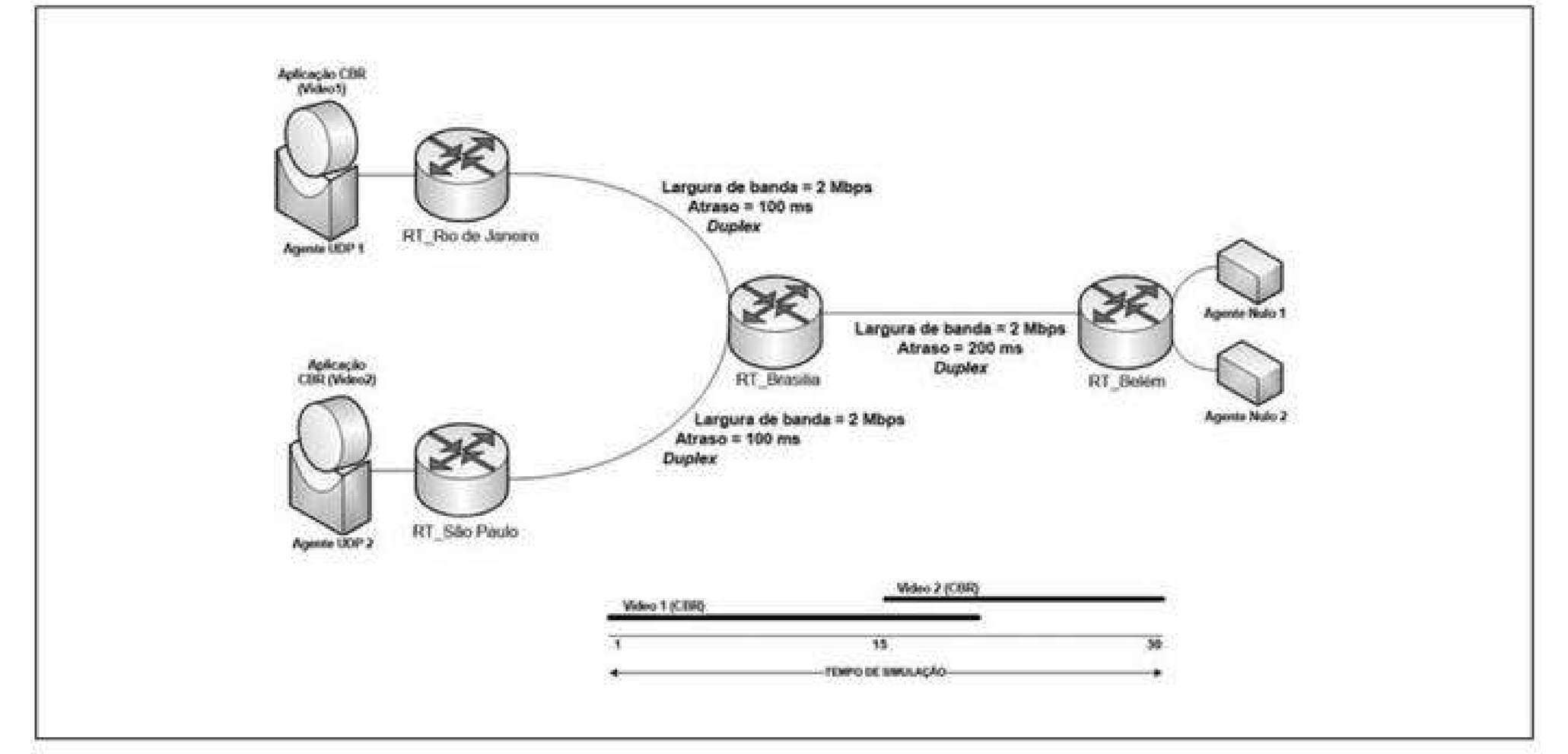


**Figura 7.6** Planejamento da simulação.

## Escalonamento dos fluxos

O escalonamento dos fluxos trata da definição dos instantes de início e fim na transmissão de cada fluxo. A interseção desses períodos de transmissão, em uma linha de tempo, definirá os pontos de maior ou menor atividade no sistema, caracterizando, inclusive, a existência de gargalos. Na Figura 7.6, o intervalo entre os instantes 15 e 20 da simulação caracteriza o ponto crítico uma vez que o somatório dos fluxos nesse intervalo equivale a uma vazão de 3 Mbps, que não é suportada em nenhum dos enlaces do cenário.

## Previsão de adversidades

Para criar uma dinâmica maior na simulação é importante inserir situações adversas que alterem o rumo natural das coisas. Exemplo disso são as quedas de enlaces que irão gerar interrupção ou mesmo redirecionamento dos fluxos, afetando direta ou indiretamente o resultado de diversas métricas utilizadas para se analisar o desempenho do sistema.

```
proc cria_conexao_cbr { origem destino inicio fim taxa tamanho_pacote classe tempomudataxa novataxa } {
global ns
set udp0 [new Agent/UDP]
set src [new Application/Traffic/CBR]
$udp0 set packetSize_$tamanho_pacote
$udp0 set class_$classe
$src set rate_$taxa
set sink [new Agent/Null]
$ns attach-agent $origem $udp0
$src attach-agent $udp0
$ns attach-agent $destino $sink
$ns connect $udp0 $sink
$ns at $inicio "$src start"
if {$tempomudataxa !=0} {
  set comando "\$ns at $tempomudataxa\ "\$src set rate_$novataxa\""
  eval $comando
}
$ns at $fim "$src stop"
return $udp0
}

set cbr_Video1 [cria_conexao_cbr$RT_RiodeJaneiro $RT_Belem 1 20 1Mb 200 1 0 0Mb]
```

**Figura 7.7** Codificação do tráfego no NS-2.

Para minimizar a quantidade de código utilizado nas etapas anteriormente descritas, uma opção é agregá-los através de funções conforme o exemplo mostrado na Figura 7.7.

# Executando e Analisando a Simulação

# 8

Para grande parte dos usuários do NS-2 este é um momento crítico, uma vez que o simulador produz como resultado apenas um arquivo-texto chamado *trace file*. O arquivo de trace é organizado cronologicamente e possui uma disposição padronizada que contém, dentre outros, o evento que foi executado (s-send; r-receive; e-enqueue; d-drop etc.), o instante de simulação onde o evento foi iniciado, a origem, o destino, o tamanho do pacote, os protocolos e uma identificação única, conforme ilustrado na Figura 8.1.

| event | time | from node | to node | pkt type | pkt size | flags | fid | src addr | dst addr | seq num | pkt id |
|---|---|---|---|---|---|---|---|---|---|---|---|

```
r : receive (at to_node)
+ : enqueue (at queue)          src_addr : node.port (3.0)
- : dequeue (at queue)          dst_addr : node.port (0.0)
d : drop    (at queue)

r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

**Figura 8.1** Arquivo de trace.

A *priori* o usuário deve buscar por outras contribuições para gerar gráficos mais elaborados que o ajudarão no processo de análise. Apesar da existência de uma ferramenta de apoio chamada *XGraph*, a obtenção dos gráficos não é automática e nem sempre trivial. Duas dicas importantes quando da importação dos dados, quer para Microsoft Excel, quer para a planilha do Open Office: a primeira é sempre substituir pontos por vírgulas nos arquivos, pois o sistema americano é diferente do brasileiro. A segunda é escolher a opção de gráfico conhecida como dispersão tomando como eixo das abscissas ($x$) o tempo de simulação, e como eixo das ordenadas ($y$), a métrica que se está pesquisando (vazão, atraso, *jitter* etc.).

Programas escritos em TCL (*Tool Command Language*) que leem o arquivo trace e o convertem a um formato pronto para ser importado para a geração dos gráficos foram desenvolvidos pelos autores deste livro e podem ser obtidos na área do leitor.

## 8.1 *Escrevendo a Simulação*

Concluída a etapa de planejamento, deve-se iniciar o mapeamento desse planejamento em um código de simulação. Isso deve ser feito na linguagem TCL. Um código exemplo será explicado por trechos no Quadro 8.1. Para melhor referência do leitor, todos os textos sublinhados no código representam mnemônicas (variáveis) e, portanto, podem ser mudados conforme opção do usuário.

Por exemplo, no item (10) o nome atribuído à aplicação foi "src". Todavia, se o usuário assim o desejasse poderia mudar esse nome para "video1", desde que o fizesse em todas as ocorrências de "src".

**QUADRO 8.1** Código exemplo de uma simulação

| Procedimentos | Código TCL | Comentários |
|---|---|---|
| (1) Sempre requerido. | **set ns [new Simulator]** | # Procedimento para instanciamento do simulador NS-2. |
| (2) Usado para permitir a reprogramação de rotas da rede em caso de queda de enlace. | **$ns rtproto DV** | # Uso de roteamento dinâmico (*Distance Vector*). Um protocolo de roteamento baseado em *Distance Vector*, baseia-se no algoritmo de Bellman-Ford para cálculo das rotas e considera as informações dos roteadores vizinhos para isso. |
| (3) Cores diferentes para fluxos diferentes. | **$ns color 1 Blue**<br>**$ns color 2 Red**<br>**$ns color 3 Yellow** | # Define cores para os fluxos de dados. É útil na visualização da simulação com o NAM (*Network Animator*). |
| (4) Cria o arquivo de trace do *Network Animator* e o associa a uma variável. | **set nf [open animacao.nam w]** | # Cria um arquivo externo chamado de "animação.nam" e o associa à variável nf (*Nam File*). |
| | **$ns namtrace-all $nf** | # Considera que todas as ocorrências (eventos) serão gravadas no arquivo "nf" em um padrão reconhecido pelo NAM. |

*(continua)*

| | | |
|---|---|---|
| (5) Cria o arquivo de trace da simulação e o associa a uma variável. | **set nt [open trace.tr w]** | # Cria um arquivo externo chamado de "trace.tr" e o associa à variável nt (*Nam Trace*). |
| | **$ns trace-all $nt** | # Considera que todas as ocorrências (eventos) serão gravadas no arquivo "nt". |
| (6) Procedimento para encerrar a simulação. | **proc encerrar {} {** | # Procedimento Encerrar. |
| | **global ns nf nt** | # Define variáveis de escopo global. |
| | **$ns flush-trace** | # Encerra a gravação nos arquivos de trace. |
| | **close $nf**<br>**close $nt** | # Fecha os arquivos. |
| | **exec nam animacao.nam &** | # Executa o programa externo NAM, passando como parâmetro o nome do arquivo onde o trace de animação foi gravado (animação.nam). |
| | **exit 0 }** | # Encerra a simulação. |
| (7) Criação dos nós. | **set rt_RiodeJaneiro [$ns node];**<br>**set rt_SaoPaulo [$ns node];**<br>**set rt_Brasilia [$ns node];**<br>**set rt_Belem [$ns node];** | # Cria os nós roteadores. |
| (8) Criação de legendas. | **$ns at 0.0 "$rt_RiodeJaneiro label rt_RiodeJaneiro"**<br>**$ns at 0.0 "$rt_SaoPaulo label rt_SaoPaulo"**<br>**$ns at 0.0 "$rt_Brasilia label rt_Brasilia"**<br>**$ns at 0.0 "$rt_Belem label rt_Belem"** | # Associa o nome dos nós a uma legenda que será exibida por ocasião da execução do NAM. |
| (9) Criação de enlaces. | **$ns duplex-link $rt_RiodeJaneiro $rt_Brasilia 2Mb 100ms DropTail** | # Cria um enlace full duplex entre RiodeJaneiro e Brasilia com capacidade máxima de vazão de 2 Mbps e atraso de 100 ms. Associa ao buffer desse segmento a política de fila DropTail, ou seja, FIFO (*First In First Out*). |

*(continua)*

| | | |
|---|---|---|
| | **$ns duplex-link $rt_SaoPaulo $rt_Brasilia 2Mb 100ms DropTail** | # Cria um enlace *full duplex* entre SaoPaulo e Brasilia com capacidade máxima de vazão de 2 Mbps e atraso de 100 ms. Associa ao buffer desse segmento a política de fila DropTail, ou seja, FIFO (*First In First Out*). |
| | **$ns duplex-link $rt_Brasilia $rt_Belem 2Mb 200ms SFQ** | # Cria um enlace *full duplex* entre Brasilia e Belem com capacidade máxima de vazão de 2 Mbps e atraso de 100 ms. Associa ao buffer desse segmento a política de fila SFQ (*Stochastic Fairness Queuing*). |
| (10) Procedimento para gerar o tráfego. | **proc cria_conexao_cbr { origem destino inicio fim taxa tamanho_pacote classe tempo-mudataxa novataxa } {** | # Parâmetros = origem, destino, instante de início do fluxo, instante de término do fluxo, taxa, tamanho do pacote, sequencial associado à cor do fluxo que foi definida no item (3), instante de mudança de taxa {se houver}, nova taxa). |
| | **global ns** | # Define variáveis de escopo global. |
| | **set udp0 [new Agent/UDP]** | # Cria um agente (protocolo de transporte) UDP chamado de udp0. |
| | **set src [new Application/Traffic/CBR]** | # Cria uma aplicação do tipo CBR (*Constant Bit Rate*) chamada de src. |
| | **$udp0 set packetSize_ $tamanho_pacote** | # Define o tamanho do pacote. |
| | **$udp0 set class_ $classe** | # Define a cor que o NAM usará para representar o fluxo mostrada em (3). |
| | **$src set rate_ $taxa** | # Define a taxa. Normalmente expressa em Mbps. |

*(continua)*

| | | |
|---|---|---|
| | **set sink [new Agent/Null]** | # Cria um agente chamado de sink que receberá o fluxo do protocolo UDP. |
| | **$ns attach-agent $origem $udp0** | # Cria uma associação entre o protocolo de transporte criado (udp0) e o nó de origem (recebido como parâmetro no procedimento). |
| | **$src attach-agent $udp0** | # Cria uma associação entre o protocolo UDP (udp0) e a aplicação que o mesmo transportará (src). |
| | **$ns attach-agent $destino $sink** | # Cria uma associação entre o que receberá o fluxo (sink) e o nó de destino (recebido como parâmetro no procedimento). |
| | **$ns connect $udp0 $sink** | # Conecta o protocolo de transporte criado (udp0) ao agente que o receberá (sink). |
| | **$ns at $inicio "$src start"** | # Define o início da transmissão do fluxo da aplicação src. |
| | **if {$tempomudataxa != 0} {<br><br>set comando "\$ns at $tempomudataxa \"\$src set rate_$novataxa\""<br><br>eval $comando<br><br>}** | # Verifica se haverá mudança de taxa ao longo da simulação. Se isso ocorrer, procede a mudança. |
| | **$ns at $fim "$src stop"** | # Define o fim da transmissão do fluxo da aplicação src. |
| | **return $udp0 }** | # Fim do procedimento. |
| (11) Chamada ao procedimento de geração de tráfego. | **set cbr_Video1 [cria_conexao_cbr $rt_RiodeJaneiro $rt_Belem 1 20 2M 1500 1 0 0M]** | # Passa os parâmetros fornecidos ao procedimento "cria_conexao_cbr". |
| | **set cbr_Video2 [cria_conexao_cbr $rt_SaoPaulo $rt_Belem 15 30 2M 1500 2 0 0M]** | |

*(continua)*

| | | |
|---|---|---|
| (12) Encerra a simulação após 30 unidades. | **$ns at 30.0 "encerrar"** | #Chama o procedimento de encerramento após decorridas "30" unidades de simulação. |
| (13) Executa a simulação. | **$ns run** | #Executa a simulação. |

## 8.2 Executando a Simulação

Concluída a escrita da simulação, basta digitar na linha de comando:

# ns <nome_do_programa.tcl>

O NS-2 não é muito amigável no processo de depuração. Todavia, se o código foi digitado corretamente, o primeiro resultado a ser observado é a apresentação da simulação na ferramenta de animação NAM (*Network Animator*). A Figura 8.2 apresenta uma fotografia (*snapshot*) coletada no instante 16,8 da simulação. Nela, pode-se observar claramente o descarte de pacotes, tanto do fluxo proveniente do roteador do Rio de Janeiro, quanto do roteador de São Paulo. Isso ocorre porque nesse momento da simulação a soma das vazões dos fluxos provenientes do Rio de Janeiro e de São Paulo (4 Mbps) excedem a capacidade do enlace de Brasília para Belém (2 Mbps).

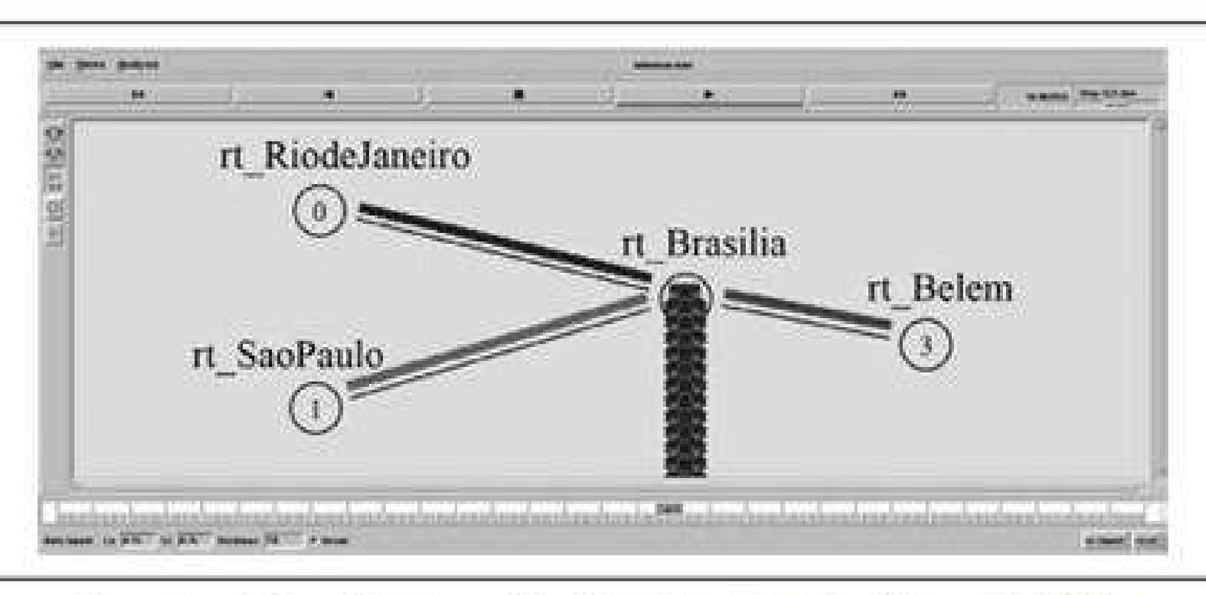


**Figura 8.2** Apresentação da simulação no NAM.

O uso de uma política de fila mais justa (SFQ – *Stochastic Fairness Queuing*) permite que diante dessa situação os recursos sejam divididos de forma equilibrada entre as duas fontes de transmissão. Os resultados podem ser observados no gráfico da Figura 8.3. Enquanto ambas as fontes encontram-se transmitindo, há uma divisão justa entre seus fluxos. Quando se encerra a transmissão do primeiro fluxo, toda largura de banda disponível é alocada ao segundo.

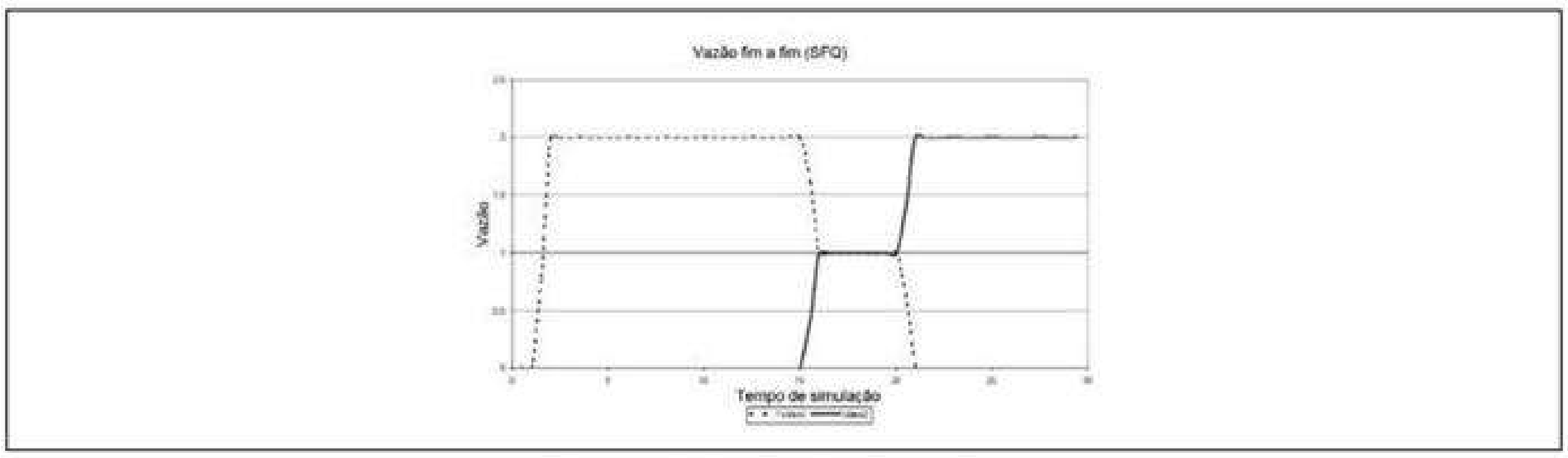


**Figura 8.3** Vazão fim a fim.

Se apenas a linha de um único fluxo (video1) for analisada, percebe-se que o desempenho cai no instante 15 da simulação. A busca pelas respostas da causa da ocorrência implica uma investigação completa dos demais elementos. Muitos analistas erram no diagnóstico de problemas de desempenho em redes reais por se aterem a um único fator, sem considerarem o que ocorre no entorno dele.

A Figura 8.4 apresenta outra medida de desempenho bastante utilizada, o atraso fim a fim. O atraso fim a fim é caracterizado como o tempo que um pacote leva para percorrer toda a rota entre o transmissor e o receptor, incluindo o tempo perdido nos *buffers* do sistema. Atrasos superiores a 300 milissegundos normalmente inviabilizam aplicações multimídia como voz e vídeo. Pode-se observar que durante o período em que os fluxos concorrem o atraso aumenta consideravelmente. Isso é causado pelo tempo extra consumido nas filas de encaminhamento dos roteadores ao longo do caminho.

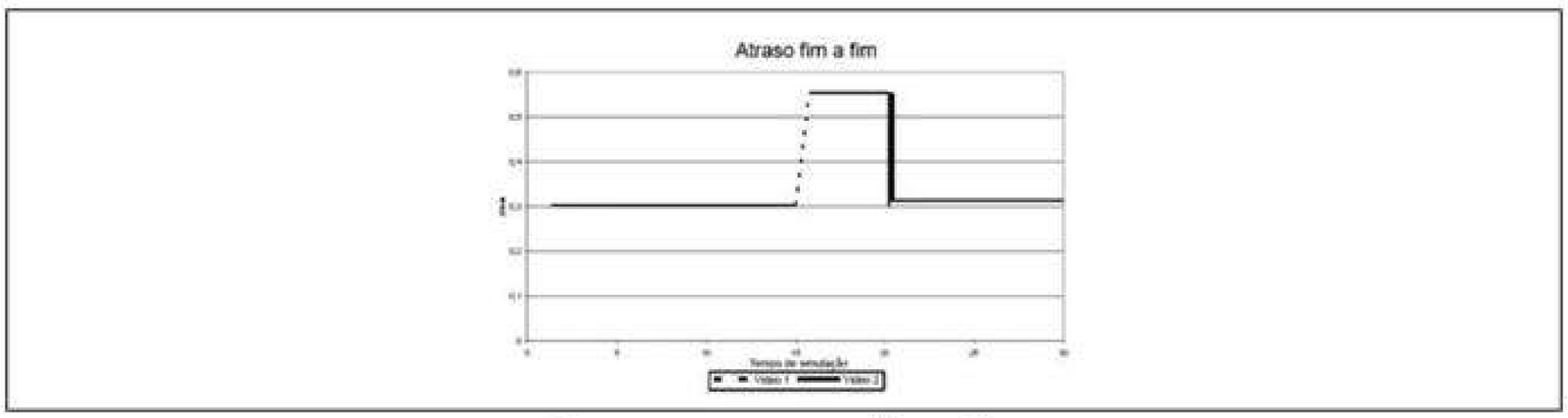


**Figura 8.4** Atraso fim a fim.

Já a Figura 8.5 apresenta o *jitter*. Trata-se de uma medida de desempenho que mede a variação estatística do atraso. Por exemplo, se o atraso de um pacote indo de São Paulo para Belém for de 10 ms e o atraso do pacote seguinte for de 15 ms, então o *jitter* será se 5 ms. Lembrando que são admitidos valores negativos para o *jitter*. Para que uma transmissão tenha um bom desempenho o *jitter* deve variar o menos possível e se manter próximo a zero. Variações grandes no *jitter* podem inviabilizar fluxos de multimídia, mesmo com uma boa vazão.

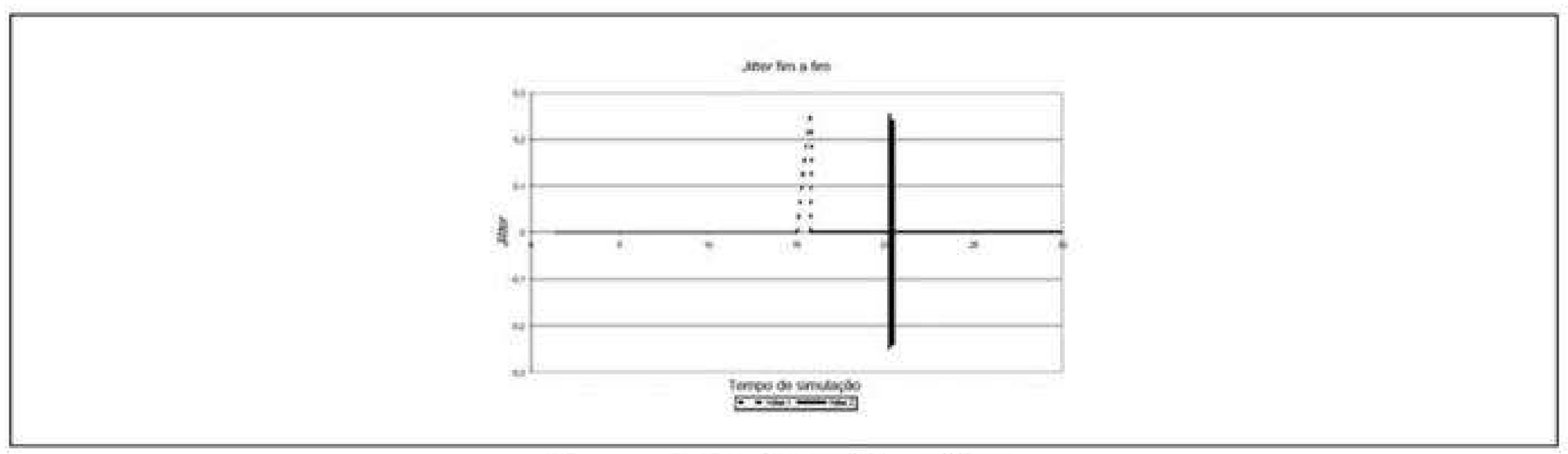


Figura 8.5 *Jitter* fim a fim.

## 8.3 Inserindo Adversidades na Simulação

Para tornar a simulação mais interessante pode-se considerar a existência de caminhos alternativos na topologia da rede. Isso permitirá que, havendo queda de enlace, o fluxo seja redirecionado. Para analisar melhor esse aspecto, será incluída uma rota alternativa entre Brasília e Belém que compreende Recife e Fortaleza, conforme ilustrado na Figura 8.6.

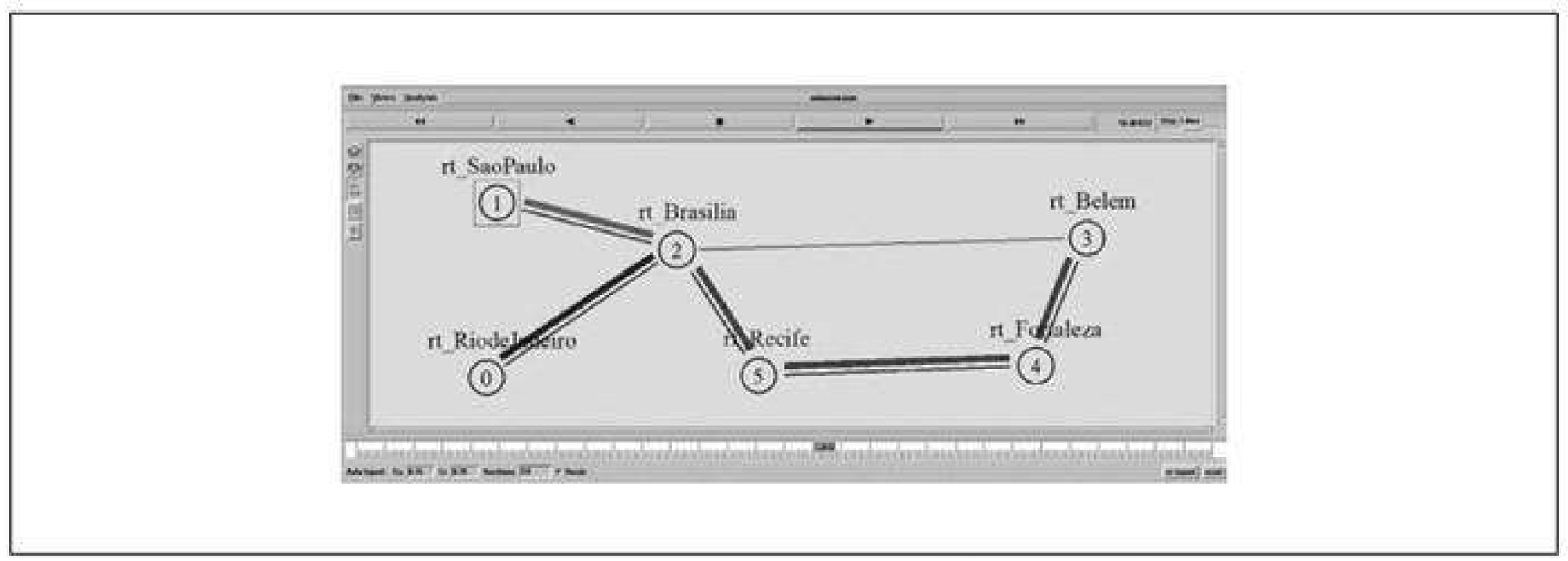


Figura 8.6 Topologia com caminho alternativo.

Em termos de programação TCL isso implica três alterações no código apresentado no Quadro 8.1.

a) Criação dos nós, onde se incluem os roteadores de Fortaleza e Recife.

| | | |
|---|---|---|
| (7) Criação dos nós. | **set rt_RiodeJaneiro [$ns node];**<br>**set rt_SaoPaulo [$ns node];**<br>**set rt_Brasilia [$ns node];**<br>**set rt_Belem [$ns node];**<br>**set rt_Fortaleza [$ns node];**<br>**set rt_Recife [$ns node];** | # Cria os nós roteadores. |

b) Criação de enlaces da rota alternativa.

| | |
|---|---|
| (9) Criação de enlaces. | **$ns duplex-link $rt_RiodeJaneiro $rt_Brasilia 2Mb 100ms DropTail**<br>**$ns duplex-link $rt_SaoPaulo $rt_Brasilia 2Mb 100ms DropTail**<br>**$ns duplex-link $rt_Brasilia $rt_Belem 2Mb 200ms SFQ**<br>**$ns duplex-link $rt_Brasilia $rt_Recife 4Mb 150ms DropTail**<br>**$ns duplex-link $rt_Recife $rt_Fortaleza 4Mb 150ms DropTail**<br>**$ns duplex-link $rt_Fortaleza $rt_Belem 4Mb 150ms DropTail** |

c) Definição dos instantes de queda e restabelecimento do enlace a ser inserido entre os itens (11) e (12) do Quadro 8.1.

| | | |
|---|---|---|
| (11.1) Indisponibilidade de enlace. | **$ns rtmodel-at 15.0 down $rt_Brasilia $rt_Belem** | # Indica uma queda de enlace entre Brasilia e Belem no instante 15 da simulação. |
| (11.2) Disponibilidade de enlace. | **$ns rtmodel-at 20.0 down $rt_Brasilia $rt_Belem** | # Indica a reativação de enlace entre Brasilia e Belem no instante 20 da simulação. |

Após os ajustes terem sido realizados e reapresentados à execução, o gráfico proveniente desse novo estado é ilustrado na Figura 8.7. Cabem aqui algumas considerações importantes que influenciarão no processo de análise.

A primeira delas diz respeito ao período de inatividade do enlace de dados entre Brasília e Belém. Isso ocorre entre os instantes 15 e 20 da simulação, fazendo com que os fluxos sejam desviados pelo trecho alternativo que compreende Recife e Fortaleza. Por ter um número de saltos maior, esse trecho não vinha sendo utilizado no encaminhamento dos pacotes. Deve-se atentar para o fato de que, durante esse período (instantes 15 a 20), ambos os fluxos, cada um com vazão de 2 Mbps, eram transmitidos simultaneamente e concorriam em uma largura de banda de 2 Mbps. O resultado se refletia em uma grande quantidade de descarte de pacotes. Com o novo cenário ocorre a queda de enlace entre Brasília e Belém justamente no período crítico da simulação, onde ambos os fluxos concorrem. Com a nova situação a rota alternativa é percorrida. É importante que se observe também a largura de banda disponível nessa nova rota que passa a ser de 4 Mbps, podendo assim comportar ambos os fluxos conforme ilustrado na Figura 8.7.

Uma medida de desempenho que pode ser utilizada para se perceber as implicações dessas mudanças é conhecida como probabilidade de bloqueio e é expressa pela relação entre o total de pacotes descartados e o total de pacotes enviados pelos fluxos:

Probabilidade de bloqueio = pacotes_descartados / pacotes_enviados

O Quadro 8.2 mostra a probabilidade de bloqueio em ambas as situações.

**QUADRO 8.2** Código exemplo de uma simulação

| Cenário | Fluxo | Enviados (E) | Descartados (D) | D/E | Bloqueio % |
|---|---|---|---|---|---|
| Sem queda de enlace e sem rota alternativa. | Rio → Belém | 22.620 | 2967 | 0,13 | 13% |
| | SãoPaulo → Belém | 17.826 | 2974 | 0,16 | 16% |
| Com queda de enlace e com rota alternativa. | Rio → Belém | 22.620 | 290 | 0,012 | 1,2% |
| | SãoPaulo → Belém | 17.841 | 57 | 0,003 | 0,3% |

Fica evidente que com o redirecionamento dos fluxos os níveis de bloqueio caíram drasticamente. Os picos, percebidos nos instantes 15 e 20, só estão presentes por conta dos ajustes ocorridos nos *buffers* (filas) dos roteadores na ocasião das mudanças de rota. Essas anomalias podem ser observadas no gráfico da Figura 8.7.

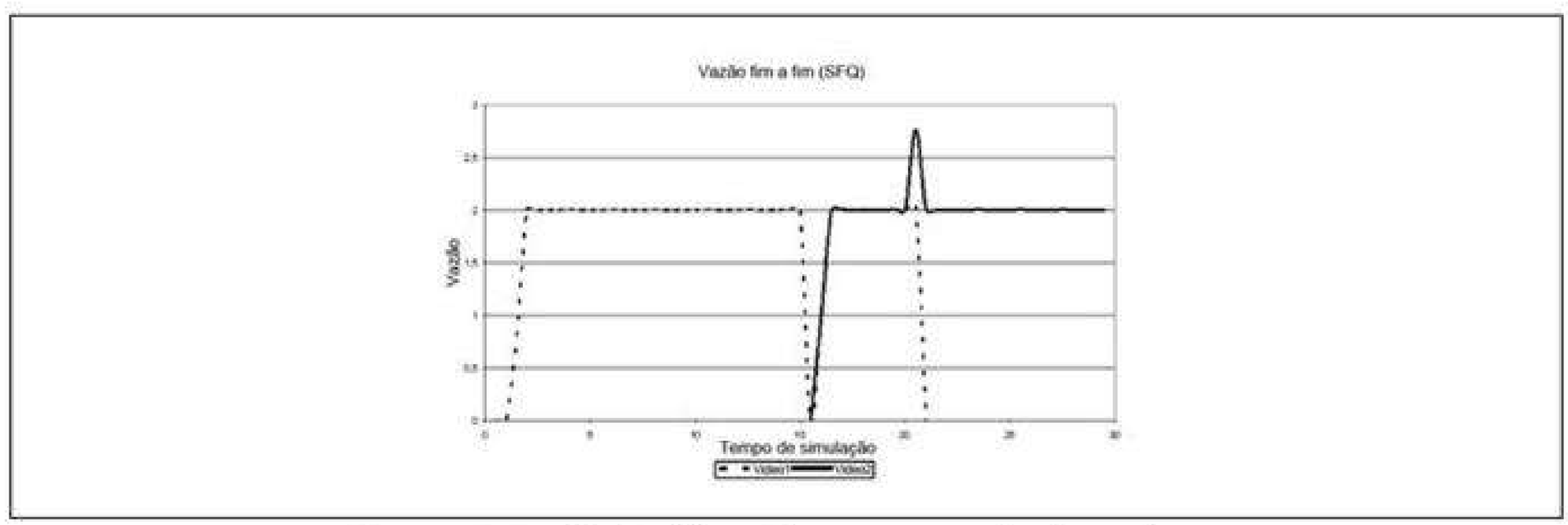


**Figura 8.7** Vazão fim a fim com queda de enlace.

## 8.4 Inserindo Randomicidade na Simulação

Outro aspecto importante para se validar estatisticamente uma simulação é criar mecanismos que permitam mudanças aleatórias a cada execução da simulação. Com isso, pode-se realizar certo número de execuções e trabalhar com uma média. Existem inúmeras variáveis no NS-2 que podem ser associadas a números aleatórios. Para que o processo fique bem claro, considerar-se-á que o instante de início da transmissão dos fluxos será sempre determinado por um número escolhido aleatoriamente entre 1 e o tempo total da simulação. O termo *seed* (semente, em português) determina um ponto fixo de início para a sequência de valores aleatórios que serão usados em uma simulação. O procedimento expresso no Quadro 8.3 realiza o cálculo desse número aleatório, e *min* equivale ao instante do início da simulação, ou seja, 0, 1, e *max* ao tempo total de simulação, ou seja, 30, se for considerado o contexto do exemplo.

**QUADRO 8.3** Procedimento para inserir um valor aleatório

| Procedimento | Código TCL |
|---|---|
| O procedimento pressupõe que dois parâmetros sejam passados na chamada do procedimento: MIN que equivale ao instante do início da simulação e MAX que corresponde ao instante final. | set nextseed 0.0<br>proc randomNumber { min max } {<br>global nextseed<br>global defaultRNG<br>$defaultRNG seed $nextseed<br>set nRNG [new RNG]<br>$nRNG next-substream<br>set num_ [new RandomVariable/Uniform]<br>$num_ set min_ $min<br>$num_ set max_ $max<br>$num_ use-rng $nRNG<br>set nextseed [expr round([$num_ value])]<br>return $nextseed<br>} |

Com isso o item (11) do Quadro 8.1 ficaria conforme a seguir apresentado.

| | | |
|---|---|---|
| (11) Chamada ao procedimento de geração de tráfego. | **set inicio1 [randomNumber 1 30]**<br><br>**set inicio2 [randomNumber 1 30]**<br><br>**set cbr_Video1 [cria_conexao_cbr $rt_RiodeJaneiro $rt_Belem $inicio1 30 2M 1500 1 0 0M]**<br><br>**set cbr_Video2 [cria_conexao_cbr $rt_SaoPaulo $rt_Belem $inicio2 30 2M 1500 2 0 0M]** | # Passa os parâmetros fornecidos ao procedimento "cria_conexao_cbr". |

O uso desse procedimento implica a geração aleatória dos instantes de início de transmissão, tanto do fluxo 1 (Rio→Belém) quanto do fluxo 2 (SãoPaulo→Belém). O mesmo princípio pode ser aplicado ao instante de queda do enlace.

# Parte IV
# Aferição


9 Experimentação ou Aferição 115

10 Inserindo Credibilidade Estatística aos Resultados de uma Avaliação de Desempenho 123

# Experimentação ou Aferição

# 9

Muitas vezes o alto nível de abstrações usadas na avaliação de desempenho via modelagem não permite que o sistema seja retratado de forma fidedigna. Dependendo do contexto, esse grau de precisão tem importância fundamental, o que pode exigir o uso de outra técnica como a experimentação ou aferição.

## 9.1 Motivação para Aferir

A motivação pelo uso da técnica de aferição normalmente se dá quando a exigência por um resultado mais preciso é fator preponderante. Entretanto, é importante frisar que aferir implica haver disponibilidade do sistema, ou de um protótipo dele, e que a realização de um processo de medição pode eventualmente causar instabilidade. Um exemplo bastante comum ocorre quando se tem a necessidade de uma expansão iminente em rede de computadores instalada e em produção. Nesse caso, pode ser necessário que a administração da empresa em questão requeira projeções, com certo grau de confiabilidade, antes de autorizar qualquer investimento. Uma das soluções criadas para se realizar medições dentro de um critério padronizado e confiável chama-se *benchmarking*.

## 9.2 *Benchmarking*

Basicamente, realizar experimentações ou aferir um sistema de forma confiável implica instalar medidores de desempenho, que podem ser tanto de hardware quanto de software, e se utilizar de processos chamados de *benchmarks*[1] para a coleta de dados e apresentação dos resultados. Existem diversas instituições dedicadas exclusivamente a definir metodologias e soluções de aferição dentro de uma margem de confiabilidade bastante segura. Um dos exemplos é a *Standard Performance Evaluation Corporation* (SPEC). A empresa mantém uma vasta *suite* de softwares de *benchmarking* dentre os quais se destacam os CPU, *webservers*, virtualização, estações gráficas etc. Os resultados obtidos são disponibilizados no endereço http://www.spec.org e podem ser usados como parâmetro comparativo.

[1] Um *benchmark* é um ponto de referência que permite um comparativo de desempenho de acordo com um conjunto de critérios preestabelecidos.

No exemplo da rede, citado anteriormente, uma possível solução seria a instalação de um programa de captura de pacotes (*sniffer*) em pontos estratégicos para coletar dados dos protocolos que por ali trafegam. Equipamentos mais modernos, como os *switches* gerenciáveis, já oferecem algumas dessas funcionalidades, trazendo inclusive ferramentas que facilitam bastante a vida dos administradores de rede. Todavia, um software de *benchmarking* vai além, pois normalmente apresenta um comparativo do resultado obtido com medições já realizadas e armazenadas em um banco de dados para que se tenha uma referência daquilo que foi mensurado.

As interpretações das informações obtidas nesse tipo de medição costumam ser extremamente úteis às instituições, principalmente por refletirem um cenário real. Com base nelas, pode-se, por exemplo, aplicar políticas diferenciadas (QoS), aumentando ou restringindo recursos de grupos de usuários conforme se faça necessário. Um grande desafio ao se medir o desempenho usando a técnica de aferição é minimizar os impactos que eventualmente o próprio mecanismo de medição venha a causar no sistema, distorcendo por vezes o resultado. Se o objetivo é medir o atraso fim a fim em uma rede de computadores e o dispositivo de medição introduzir algum atraso nessa rede, então é importante analisar se esse incremento não irá comprometer o resultado como um todo.

Um exemplo de software de *benchmark* gratuito e bastante popular é o *PC Wizard* 2010, que pode ser obtido no endereço www.cpuid.com. Nele, as informações de seu computador são coletadas e, opcionalmente, comparadas com a de outros computadores. A Figura 9.1 mostra o resultado de um *benchmark* de CPU realizado em um computador com processador Intel Pentium E 2140 de 1600 MHz. Os números resultantes, apresentados na área superior, são expressos em milhões de instruções por segundo (MIPS) e de milhões de operações de ponto flutuante por segundo (MFLOP). O botão *Compare With* pode ser usado para uma comparação com dados armazenados no banco de dados. A Figura 9.2 mostra uma comparação com um processador Intel Core 2 Quad Q9550. Ainda é possível inserir um certo grau de transparência para sobrepor as imagens de ambos, permitindo uma melhor percepção da diferença.

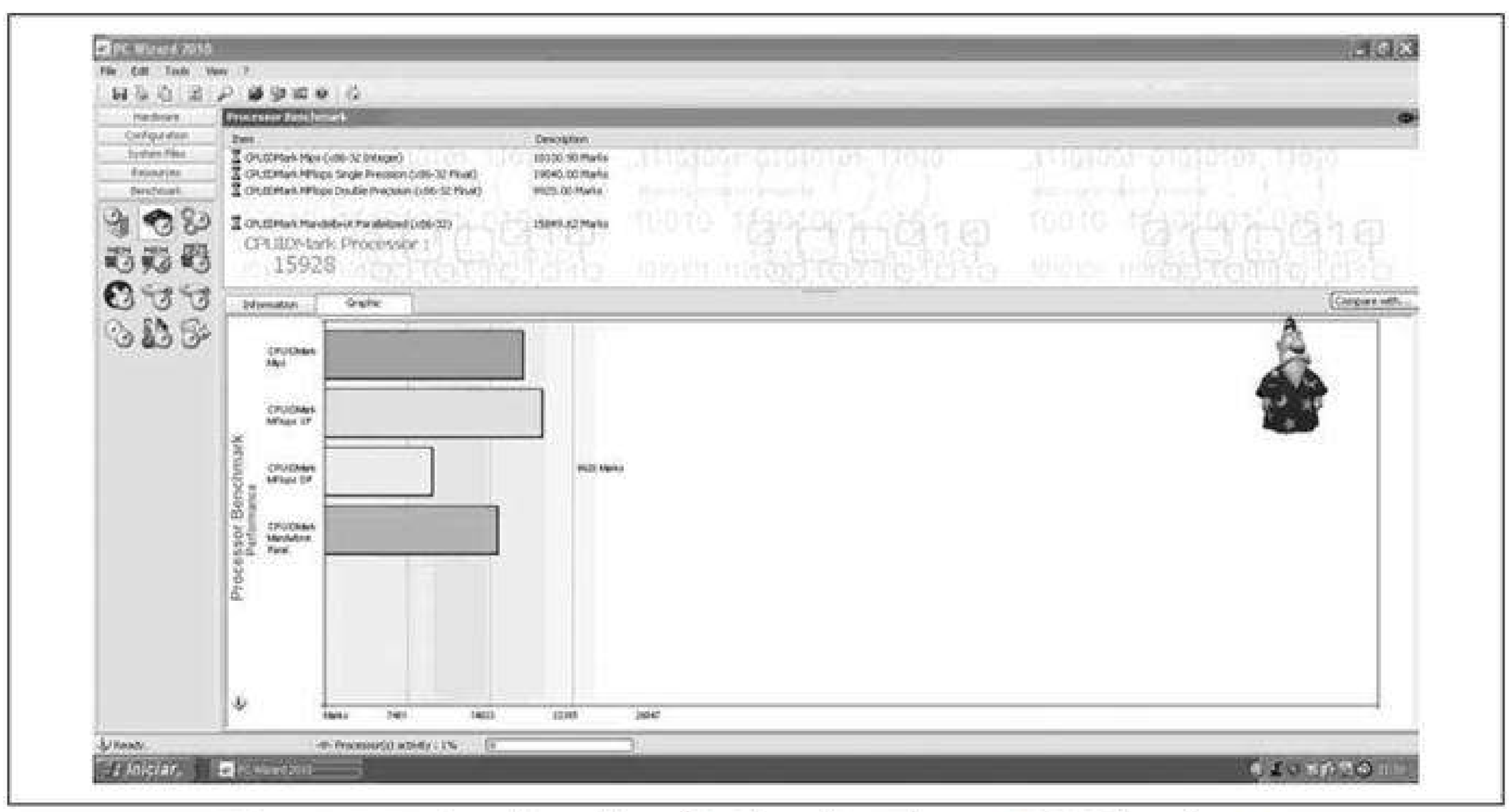


**Figura 9.1** *Benchmark* realizado pelo software *PC Wizard* 2010.

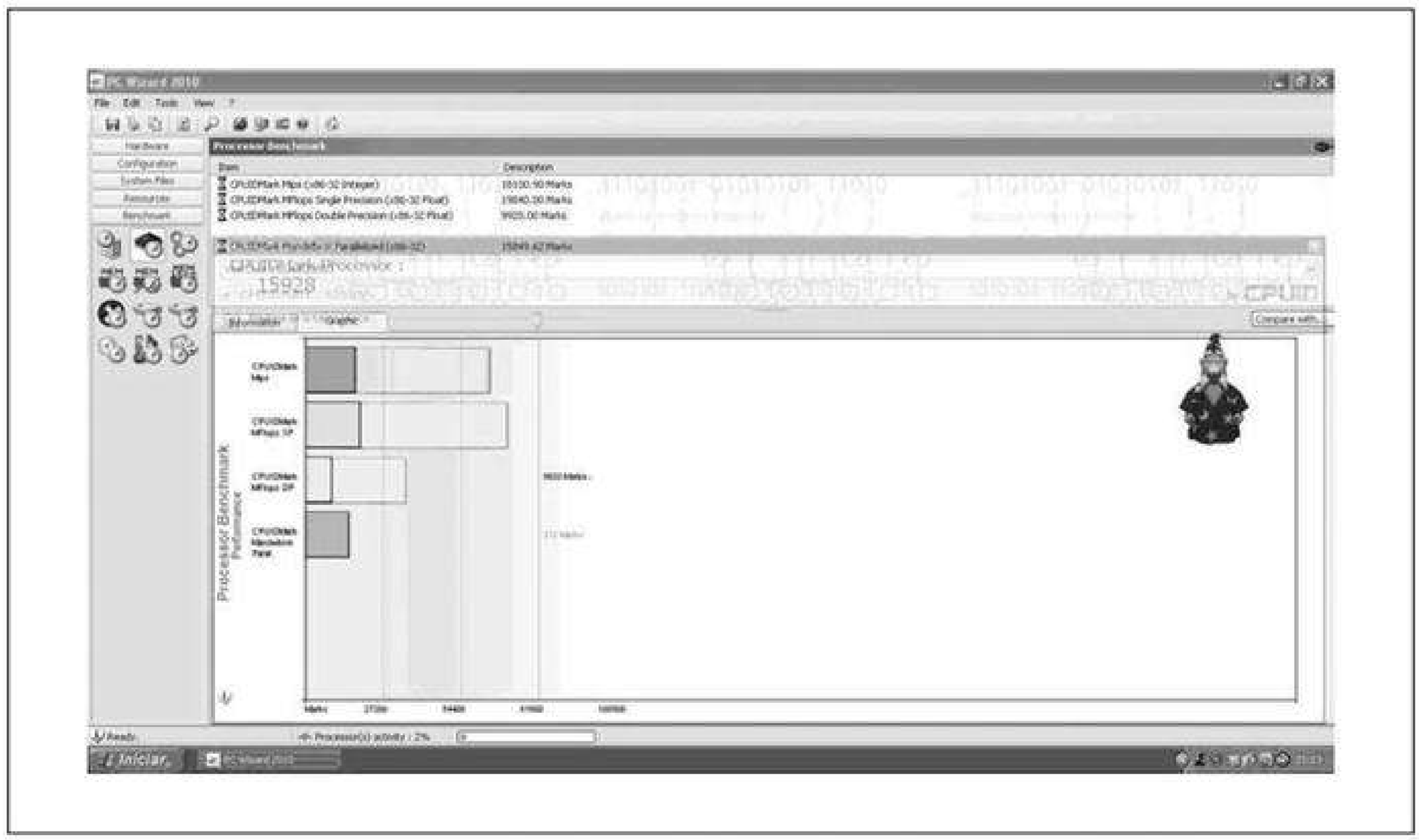


**Figura 9.2** Comparativo de resultados pelo software *PC Wizard* 2010.

Além dos testes de desempenho de CPU, o *PC Wizard* disponibiliza testes de memória, gráficos, rede etc. Outro exemplo de software de medição bastante utilizado no meio acadêmico é o IPERF.

## 9.3 Utilização do IPERF em Aferição

Muitas vezes uma solução simples, baseada em aferição, pode ajudar no diagnóstico de problemas do cotidiano. A seguir será contextualizado um exemplo usando o software IPERF com o objetivo de ilustrar uma dessas situações. Cabe ressaltar que a ferramenta IPERF escrita em C++ é gratuita e está disponível para download no *Source/Forge*. Foi desenvolvida pelo Laboratório Nacional para Pesquisas em Redes Aplicadas (NLANR) e pelo Time de Suporte a Aplicações Distribuídas (DAST) como uma alternativa para medir o desempenho máximo de largura de banda tanto com TCP quanto com UDP. Dentre os resultados reportados pelo IPERF estão a largura de banda, atraso, *jitter* e probabilidade de bloqueio. Um complemento chamado de JPERF (Figura 9.3) agrega funcionalidades como geração de gráficos e uma interface gráfica que facilita o processo.

Endereço para download do IPERF: http://sourceforge.net/projects/iperf/

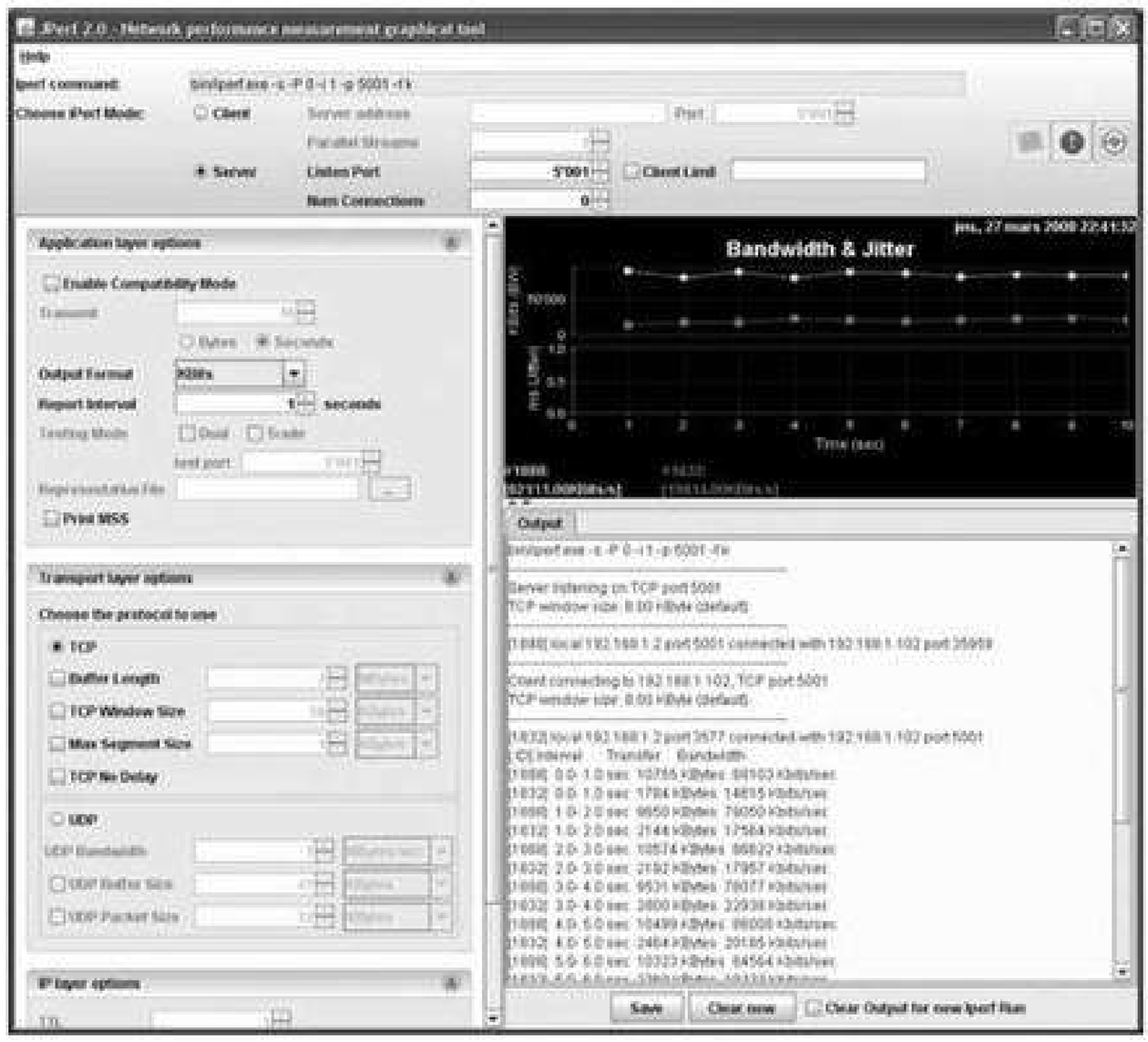


**Figura 9.3** Interface do software JPERF.

## 9.3.1 Contextualização do Cenário

Um dos executivos de uma empresa denominada "Fictícia S.A." solicitou ao Centro de Tecnologia da Informação e Comunicação (CTIC) a realização de um estudo de custo/benefício para aquisição de um sistema de rede sem fio que iluminasse[2] toda a área física da instituição. O investimento seria significativo uma vez que mais de 80 equipamentos estavam sendo estimados.

Na reta final, três fornecedores disputavam a venda de seus produtos. Um deles propunha roteadores sem fio padrão IEEE 802.11b a um custo X. O outro oferecia roteadores do mesmo fabricante, no padrão IEEE 802.11g, a um custo X + 10%. Já o terceiro oferecia seus equipamentos no padrão IEEE 802.11n, a um custo 2,5X.

Com o objetivo de agregar informações técnicas mais consistentes ao relatório final, o CTIC optou pela realização de um teste de desempenho com os equipamentos de todos os fornecedores. Para tanto, foi utilizado no processo de medição o software IPERF, que inseriu carga no enlace e mediu vários parâmetros de desempenho. Cabe ressaltar que as medições foram realizadas durante um período de 15 minutos sob as mesmas circunstâncias, mas sem o rigor técnico que impõe o tratamento estatístico/documentativo mais apurado, exigido em um trabalho profissional. A ilustração e parametrização do teste podem ser observadas, respectivamente, na Figura 9.4 e no Quadro 9.1.

[2] O termo iluminar é usado para fazer referência à área coberta com sinal de radiofrequência dentro de um nível mínimo de qualidade preestabelecido.

**QUADRO 9.1** Configuração dos hosts

| ***Hosts*** | |
|---|---|
| **Servidor IP 10.0.0.69/8** | **Cliente IP 10.0.0.42/8** |
| **iperf -s -u -i (intervalo de apresentação) > arquivo.txt** | **iperf -c (IP do servidor) -u -b 54 m -t (tempo para medição)** |
| -s = servidor<br>-u = udp<br>-i = intervalo no qual os dados lidos são exibidos em segundos (10) | -c = cliente<br>-u = udp<br>-b = velocidade (carga) a ser injetada no link<br>-m = formato em megabits<br>-t = tempo de medição em segundos (900) |

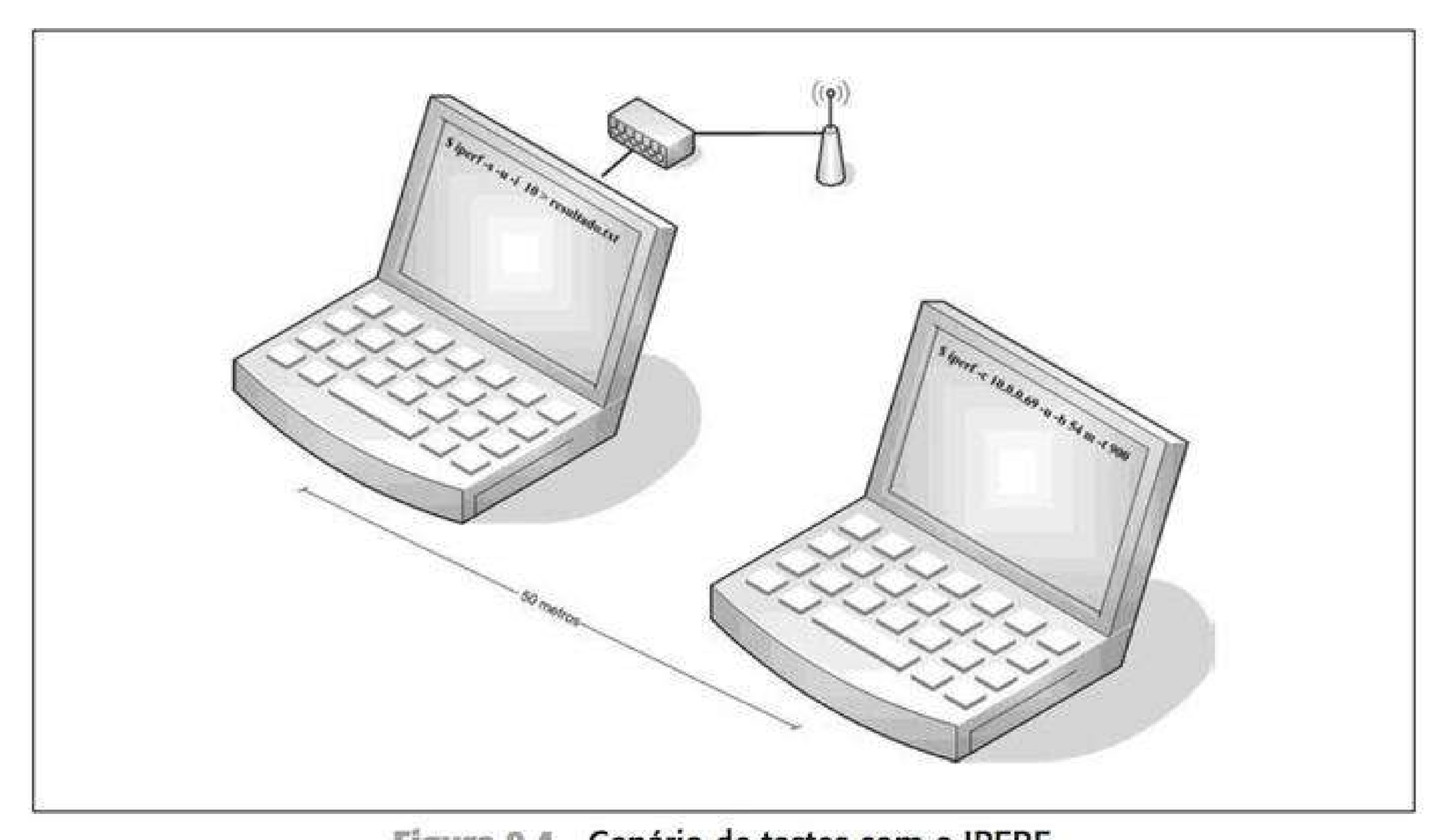


**Figura 9.4** Cenário de testes com o IPERF.

A proposta foi injetar carga compatível com a velocidade nominal[3] de cada equipamento. Sendo assim, foram usadas taxas de 11, 54 e 300 megabits por segundo (Mbps), de acordo com o equipamento testado. O objetivo era analisar o desempenho em todos os equipamentos diante de uma situação real. Ao final das medições o IPERF apresenta os resultados conforme ilustração da Figura 9.5. Faz-se ainda necessário que esse resultado seja importado, organizado e tratado para se chegar à etapa de geração dos gráficos, o que pode ser feito com qualquer software de planilha eletrônica como *open office*, da Oracle.

[3] A velocidade dita nominal normalmente é aquela anunciada pelo fabricante na caixa do produto e dificilmente se equipara à velocidade efetiva do mesmo, que é a obtida em ambientes reais.

```
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 8.00 KByte (default)
------------------------------------------------------------
[1928] local 192.168.254.1 port 5001 connected with 192.168.254.2 port 1686
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928]  0.0-10.0 sec  17.7 MBytes  14.8 Mbits/sec  1.154 ms 1937331036/12607 (1.5e+007%)
[1928] 10.0-20.0 sec  20.4 MBytes  17.2 Mbits/sec  1.120 ms    6/14593 (0.041%)
[1928] 20.0-30.0 sec  21.9 MBytes  18.4 Mbits/sec  0.841 ms   19/15651 (0.12%)
[1928] 30.0-40.0 sec  22.1 MBytes  18.5 Mbits/sec  0.867 ms   16/15788 (0.1%)
[1928] 40.0-50.0 sec  22.1 MBytes  18.5 Mbits/sec  1.302 ms   21/15781 (0.13%)
[1928] 50.0-60.0 sec  18.7 MBytes  15.7 Mbits/sec  1.262 ms   10/13337 (0.075%)
[1928] 60.0-70.0 sec  17.7 MBytes  14.8 Mbits/sec  1.057 ms    7/12606 (0.056%)
[1928] 70.0-80.0 sec  20.0 MBytes  16.8 Mbits/sec  1.386 ms    9/14292 (0.063%)
[1928] 80.0-90.0 sec  17.5 MBytes  14.7 Mbits/sec  1.482 ms    3/12515 (0.024%)
[1928] 90.0-100.0 sec  19.5 MBytes  16.3 Mbits/sec  1.387 ms    4/13892 (0.029%)
[1928] 100.0-110.0 sec  20.6 MBytes  17.3 Mbits/sec  1.078 ms    6/14691 (0.041%)
[1928] 110.0-120.0 sec  22.1 MBytes  18.5 Mbits/sec  1.498 ms   11/15777 (0.07%)
[1928] 120.0-130.0 sec  21.5 MBytes  18.0 Mbits/sec  1.116 ms    9/15340 (0.059%)
[1928] 130.0-140.0 sec  20.8 MBytes  17.4 Mbits/sec  1.273 ms    6/14838 (0.04%)
```

**Figura 9.5** Resultado das medições realizadas com o IPERF.

Após os dados obtidos terem sido devidamente organizados, pode-se apresentá-los graficamente. Os gráficos das Figuras 9.6 e 9.7 mostram, respectivamente, os resultados das medições de vazão e *jitter* para ambos os equipamentos.

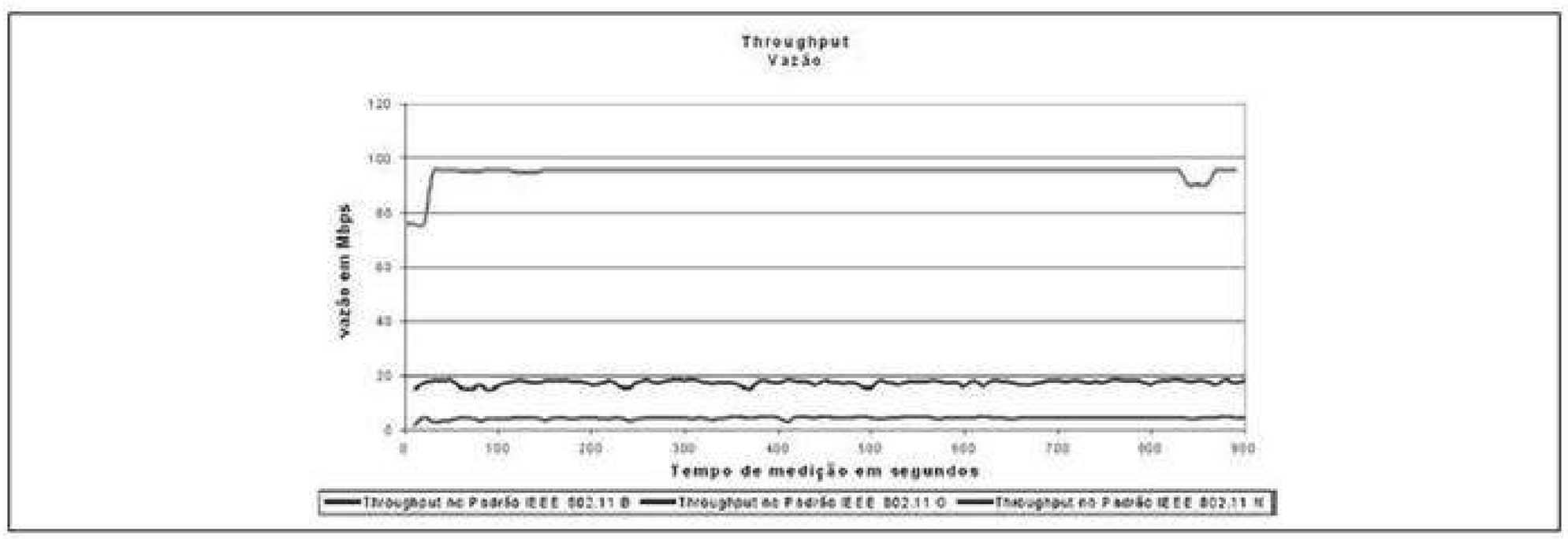


**Figura 9.6** Vazão fim a fim obtida durante 15 minutos de medição.

O gráfico da Figura 9.6 mostra claramente que as taxas nominais informadas pelos fabricantes não são alcançadas em um ambiente real em função das diversas variáveis locais de interferência. Os valores nominais são normalmente obtidos em ambientes controlados (laboratórios).

O ponto de acesso padrão IEEE 802.11b, que tem como velocidade nominal 11 Mbps, alcançou apenas a velocidade efetiva média de 4,60 Mbps, ou seja, aproximadamente 41,81% da velocidade nominal, enquanto o ponto de acesso padrão IEEE 802.11g, que tem como velocidade nominal 54 Mbps, alcançou uma velocidade efetiva média de 17,59 Mbps, o que equivale a aproximadamente 32,57% da velocidade nominal. Já o ponto de acesso padrão IEEE 802.11n, que tem como velocidade nominal 300 Mbps,[4] alcançou uma velocidade efetiva média de 94 Mbps, aproximadamente 31,3% da velocidade nominal. Cabe ressaltar que os equipamentos

[4] A proposta do padrão IEEE 802.11n é operar a uma taxa de transmissão em torno de 74 Mbps. Todavia, alguns fabricantes garantem taxas nominais maiores se apenas os seus equipamentos forem envolvidos no processo. Um desses fabricantes é a TPLINK, que foi usada nesta experimentação.

vendidos atualmente no mercado como IEEE 802.11n são baseados em uma especificação preliminar, uma vez que o padrão definitivo ainda se encontra em fase de discussão.

Em todos os casos não há como definir um comportamento padrão, pois cada ambiente impõe suas variáveis de interferência[5] e, portanto, tem sua particularidade. Apenas um trabalho de medição *in loco* pode revelar um quadro mais realista como retratado neste exemplo. Esse processo faz parte de um conjunto de procedimentos chamados de *site survey*, que inclui a visita de um consultor de redes para determinar o hardware, o posicionamento dos pontos de acesso, a área de cobertura, enfim, a viabilidade técnica e o custo estimado do projeto.

O *jitter*, apresentado no gráfico da Figura 9.7, indica a variação estatística do atraso e, portanto, pode ser tanto positivo quanto negativo. Se o atraso medido entre um ponto A e um ponto B for de 10 milissegundos e o atraso seguinte for de 4 milissegundos, então o *jitter* será negativo, ou seja, de -6 milissegundos. Aqui, ao contrário da vazão, quanto mais o gráfico se aproximar de zero, melhor o desempenho. Uma ocorrência muito instável de *jitter*, mesmo com uma boa vazão, pode comprometer aplicações de multimídia. Por isso o contexto do cenário sempre deve ser considerado. Se, por exemplo, a empresa em questão for de publicidade e propaganda, onde normalmente se abusa do tráfego multimídia, uma variação grande de *jitter* poderia comprometer, significativamente, os serviços.

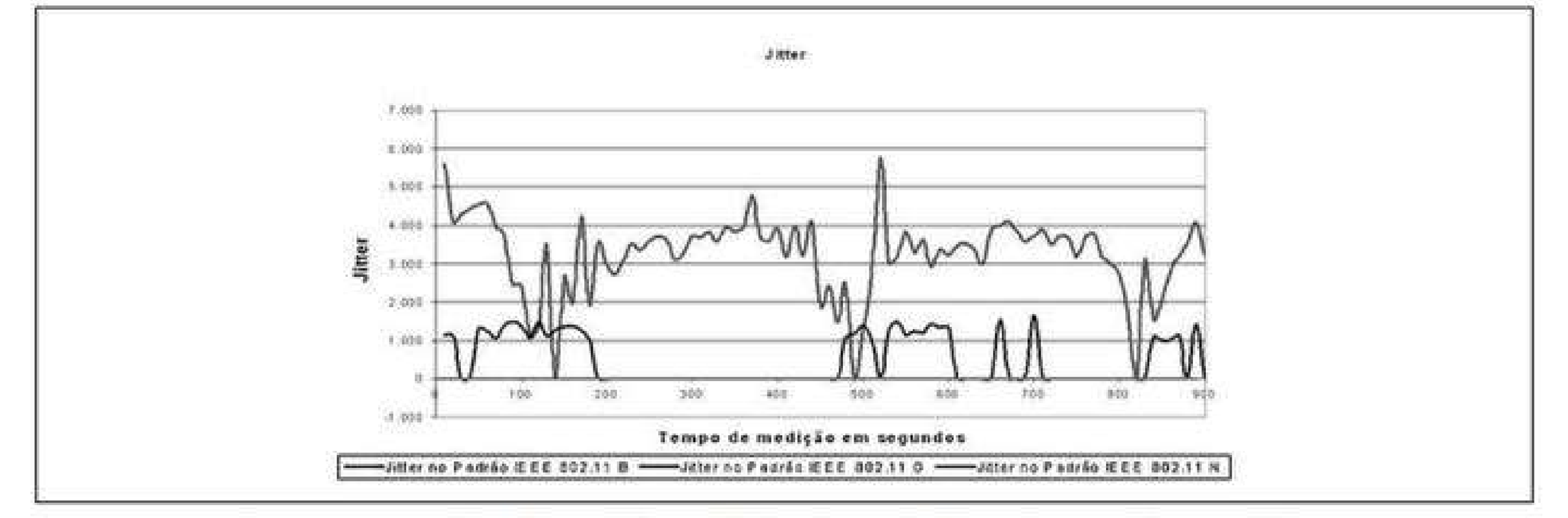


**Figura 9.7** *Jitter* fim a fim obtido durante 15 minutos de medição.

Certamente a decisão pela aquisição de um ou outro equipamento ainda dependeria de uma série de fatores. Todavia, a informação coletada e apresentada servirá como subsídio para que a escolha tenha um respaldo técnico consistente.

Há de se ressaltar que alguns projetistas de rede sem fio preferem medir o nível de sinal no lugar das taxas de transmissão. Isso pode ser feito com softwares gratuitos como o *Wavemon* para o Linux,[6] conforme ilustrado na Figura 9.8. Nesse caso, o nível de sinal deve ser comparado com o limiar de recepção do adaptador de rede do cliente, que normalmente é disponibilizado pelo fabricante e gira em torno de -90 dBm.

---

[5] Entenda-se por variáveis de interferência paredes, aquários, divisórias de vidro etc.

[6] Nas distribuições derivadas do Debian, o *Wavemon* pode ser instalado com o *comando apt-get install wavemon.*

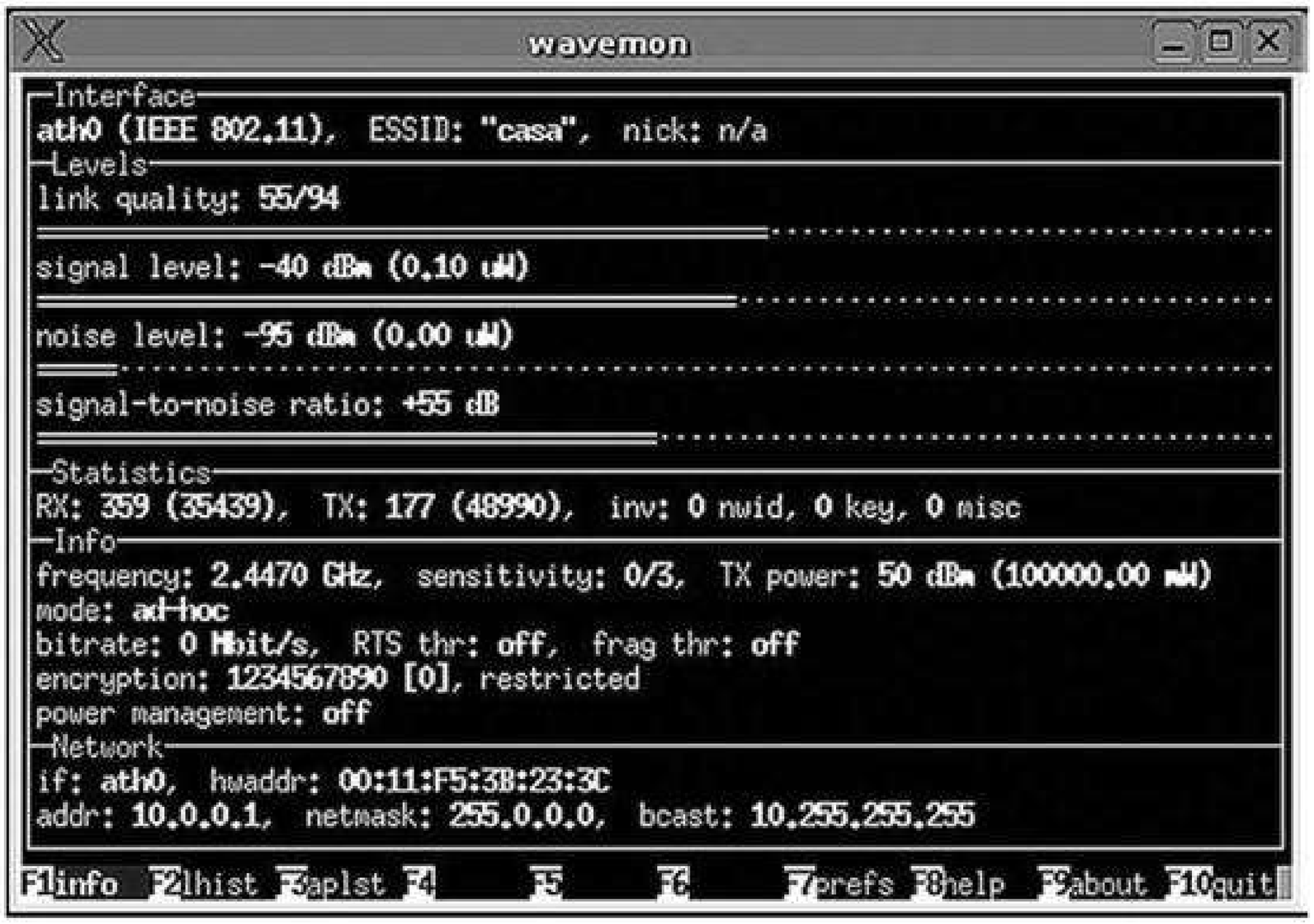


**Figura 9.8** Tela do Wavemon.

# Inserindo Credibilidade Estatística aos Resultados de uma Avaliação de Desempenho

# 10

## 10.1 Introdução

Quando se avalia o desempenho de um sistema, o tratamento estatístico aplicado à amostra é tão importante quanto o planejamento do experimento. Um dos cuidados que se deve ter é aguardar até que o sistema atinja certo grau de estabilidade. Esse intervalo de espera, chamado de *warm-up*, normalmente compreende cerca de 10% do tempo inicial do processo. Por isso, os resultados obtidos nos instantes iniciais não devem ser considerados na análise.

Outro item que contribui para a credibilidade dos resultados é a aplicação do chamado intervalo de confiança. O Intervalo de Confiança (IC), ou estimativa intervalar, define uma amplitude, um intervalo de valores com alta probabilidade de conter a verdadeira média da população amostral.

Um intervalo de confiança está sempre associado a um grau de confiança. Se, por exemplo, o valor de 95% for usado, significa que existe 95% de probabilidade de a média da população se situar entre os limites do intervalo calculado. Um IC expressa, portanto, a ideia de que se a média real estiver fora desse intervalo, as chances de a amostra representar satisfatoriamente a população são muito pequenas.

Neste capítulo será visto como calcular o Intervalo de Confiança de dados obtidos por meio de simulação, experimentação ou modelagem. O IC também será usado como parâmetro de comparação de dois ou mais sistemas que estão sendo avaliados.

## 10.2 Intervalo de Confiança da Média

Ao se realizar estudos de avaliação de desempenho, na maioria dos casos, é impossível ou muito trabalhoso obter todos os dados de todos os experimentos possíveis, isto é, da *população*. Por exemplo, um sistema *web* que atende requisições HTTP a uma taxa muito grande terá uma quantidade de dados brutos de desempenho muito grande, que consumirá muitos recursos para sua análise. Uma forma de analisar o sistema será através de um conjunto de *amostra*, que é apenas uma parte dos dados, mas representativa para toda a população.

Se a média da população já é conhecida, não é necessário o cálculo de um IC. Entretanto, o uso de amostras dos sistemas analisados permite fazer uma análise probabilística do desempenho de um sistema. Afinal, amostras são exemplos de dados que foram coletados e usados para analisar o comportamento de um sistema.

Portanto, quando se calcula um intervalo de confiança, é calculada a média das amostras para se obter uma estimativa para a média da população. O IC da média define um intervalo de valores com alta probabilidade de conter a verdadeira média da população amostral.

### *10.2.1 Conceitos Básicos*

Para o cálculo de um Intervalo de Confiança são necessários alguns dados sobre o sistema a ser analisado:

- Média $(\bar{x})$: média aritmética, que é o valor médio resultante das medidas das amostras existentes.
  Por exemplo, foram medidos 30 tempos de resposta em um sistema de transações. A média dos tempos de resposta é a soma de todos os tempos coletados dividida por 30.
- Coeficiente de confiança ou nível de significância $(\alpha)$: é o parâmetro que informa com qual confiança quer-se localizar o intervalo que contém a média das amostras.

O IC obtém a estimativa de que a média $\bar{x}$ de um sistema analisado esteja contida em um intervalo, onde a probabilidade de a média estar contida nesse intervalo seja igual a um nível de confiança desejado:

$$\text{Prob}\{c_1 < c < c_2\} = 1 - \alpha$$

onde:

$(c_1, c_2)$ é o intervalo de confiança, em que $c_1$ é o limite inferior e $c_2$ é o limite superior do intervalo;

$\alpha$ é o nível de significância;

$100(1 - \alpha)$ é o nível de confiança.

Na avaliação de sistemas computacionais, o valor geralmente utilizado para nível de confiança varia de 90% a 99%.

**Exemplo**

Uma conexão Fast Ethernet entre duas redes locais foi analisada para verificação do tempo médio de resposta de transmissão de dados entre as duas redes. O gerente da rede tinha o cálculo do intervalo de confiança do tempo de resposta de transmissão de um arquivo de 1 GB, que variava de 90,34 segundos a 93,01 segundos, dependendo do tráfego no link, com 90% de confiança. Portanto, tem-se:

$$\text{Prob}\{90{,}34 \leq m \leq 93{,}01\} = 0{,}9$$

e

$$100(1-\alpha) = 90$$
$$\alpha = 0{,}1$$

### *10.2.2 Teorema do Limite Central*

Quando se consideram amostras grandes, tem-se que a distribuição das médias amostrais será, de forma aproximada, normalmente distribuída. Essas médias tenderão a uma distribuição Normal à medida que o tamanho da amostra cresce. Pode-se ter então uma variável com uma distribuição diferente da Normal, e ao se utilizar várias amostras grandes desta distribuição para criar um histograma das médias amostrais, a forma resultante se parecerá como uma curva Normal.

O **Teorema do Limite Central** diz que a aproximação desse histograma para a distribuição Normal melhora à medida que o tamanho amostral cresce. Isso permite que, independentemente do tipo da distribuição da população, seja possível inferir dados e realizar os cálculos desejados para essa população. Portanto, para uma amostra suficientemente grande a distribuição das médias amostrais em torno da média populacional é Normal com desvio padrão $\sigma/\sqrt{n}$, onde $\sigma$ é o desvio padrão da população e $n$ o tamanho da amostra.

Como é possível não saber o verdadeiro desvio padrão populacional $\sigma$ para grandes tamanhos amostrais, o desvio padrão amostral $s$ é uma boa estimativa de $\sigma$. Portanto, substitui-se $\sigma$ por $s$ de modo que seja possível calcular o erro padrão como $s/\sqrt{n}$.

No caso do cálculo do Intervalo de Confiança, deseja-se conhecer o limite inferior e o limite superior onde a média das amostras se localiza com um certo nível de confiança. Então, um intervalo de confiança de $100\,(1-\alpha)\%$ de confiança para a média de uma população é dado por:

Limite inferior: $\bar{x} - z_{1-\alpha/2}\, s/\sqrt{n}$

Limite superior: $\bar{x} + z_{1-\alpha/2}\, s/\sqrt{n}$

onde:

- $\bar{x}$ é a média das amostras;
- $s$ é o desvio padrão das amostras;
- $n$ é o tamanho da amostra;
- $Z_{1-\alpha/2}$ é um $(1-\alpha/2)$ quantil[1] de uma Normal. Significa que representa o número de desvios padrões que se estendem da média de uma distribuição Normal necessário para conter $1-\alpha/2$ da área.

O valor de $Z_{1-\alpha/2}$ pode ser encontrado na tabela da Distribuição Normal Padronizada (Anexo A). Na tabela, deve-se procurar a linha e a coluna com o valor mais aproximado para $z$ de $1-\alpha/2$.

[1] Quantis são pontos estabelecidos em intervalos regulares a partir da função distribuição acumulada de uma variável aleatória. Dividem os dados ordenados em $n$ subconjuntos de dados de dimensão essencialmente igual.

Por exemplo, calculando $z$ para um nível de confiança de 90%, tem-se:

$$1-\frac{\alpha}{2}=1-\frac{0,1}{2}=0,95 \text{ e } z=1,645$$

Outro exemplo, calculando $z$ para um nível de confiança de 95%, tem-se:

$$1-\frac{\alpha}{2}=1-\frac{0,05}{2}=0,975 \text{ e } z=1,960$$

O Intervalo de Confiança da média pode ser também escrito no formato de um intervalo, diretamente:

$$(\bar{x} - z_{1-\alpha/2}s/\sqrt{n}, \bar{x} + z_{1-\alpha/2}s/\sqrt{n})$$

Ou, em outro formato mais compacto:

$$\bar{x} \pm z_{1-\alpha/2}s/\sqrt{n}$$

**Exemplo 1**

Trinta e quatro amostras de tempo de resposta foram coletadas de uma rede de computadores em diferentes momentos. O conjunto M representa as amostras coletadas (milissegundos):

M = {840, 860, 850, 840, 860, 835, 1100, 500, 825, 833, 830, 830, 835, 840, 845, 825, 844, 845, 2000, 820, 840, 843, 844, 845, 823, 844, 843, 855, 851, 811, 847, 850, 834, 849}

Dos valores coletados, calculam-se os dados necessários para encontrar o IC (arrendondando para duas casas decimais):

- Número de amostras: $n = 34$
- Média: 816,41
- Desvio padrão: 164,26

**Cálculo do IC com 90% de confiança**

Considerando o nível de confiança igual a 90%, para calcular o nível de significância, tem-se que:

$$NC = 100(1-\alpha)$$
$$90 = 100(1-\alpha)$$
$$\alpha = 0,1$$

*(continua)*

Segundo a tabela da distribuição Normal, para $1 - \alpha/2 = 0{,}95$, o valor do parâmetro $z$ é igual a 1,645.

Portanto,

$$IC = (\bar{x} - z_{1-\alpha/2} s/\sqrt{n}, \bar{x} + z_{1-\alpha/2} s/\sqrt{n})$$

$$IC = (816{,}41 \pm 1{,}645 \times 164{,}26/\sqrt{34})$$

$$IC = (770{,}07;\ 862{,}75)$$

O resultado diz que, com 90% de certeza, a média dos valores de tempo de resposta está contida no intervalo de 770,07 a 862,75 milissegundos.

**Cálculo do IC com 95% de confiança**

Considerando o nível de confiança igual a 95%, para calcular o nível de significância, tem-se que:

$NC = 100(1 - \alpha)$
$95 = 100(1 - \alpha)$
$\alpha = 0{,}05$

Segundo a tabela da distribuição Normal, com $1 - \alpha/2 = 0{,}975$, o valor do parâmetro $z$ é igual a 1,960.

Portanto,

$$IC = (\bar{x} - z_{1-\alpha/2} s/\sqrt{n}, \bar{x} + z_{1-\alpha/2} s/\sqrt{n})$$

$$IC = (816{,}41 \pm 1{,}960 \times 164{,}26/\sqrt{34})$$

$$IC = (761{,}20;\ 871{,}63)$$

O resultado diz que, com 90% de certeza, a média dos valores de tempo de resposta está contida no intervalo de 761,20 a 871,63 milissegundos.

É importante observar que, quanto maior o nível de confiança desejado, maior o intervalo de confiança obtido.

90% de confiança: IC = (770,07; 862,75)
95% de confiança: IC = (761,20; 871,63)

**Exemplo 2**

A maioria dos aplicativos de planilha eletrônica como MS-Excel e OpenOffice possuem a função INT.CONFIANÇA que solicita, como argumentos, o nível de significância, o desvio padrão e o número de amostras.

Os valores apresentados em letras normais correspondem àqueles situados dentro do Intervalo de Confiança e os apresentados em itálico indicam os que estão fora do intervalo.

**QUADRO 10.1** Cálculo do intervalo de confiança utilizando uma planilha eletrônica

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | **Vazão em Mbps** | | **Fórmula** | **Valor** |
| 2 | 840 | **Média** | =MEDIA(B3:B35) | 816,4117647 |
| 3 | 860 | **Desvio padrão** | =DESVPAD(B3:B35) | 164,2620076 |
| 4 | 850 | **Nível de significância** | | 0,01 |
| 5 | 840 | **Número de amostras** | | 34 |
| 6 | 860 | | | |
| 7 | 835 | **Intervalo de Confiança** | =INT. CONFIANÇA(E4;E3;E6) | 46,33668118 |
| 8 | 1100 | | | |
| 9 | 500 | | | |
| 10 | 825 | | | |
| 11 | 833 | **Média – IC** | **Média** | **Média + IC** |
| 12 | 830 | 770,0750835 | 816,4117647 | 862,7484459 |

**Exemplo 3**

No gráfico da Figura 10.1 pode-se perceber a diferença quando apenas os valores contidos no Intervalo de Confiança são considerados e quando todos os valores da amostra são considerados.

O uso do critério do IC desconsidera aqueles valores de amostras que não são representativos e, portanto, podem descaracterizar o gráfico resultante.

**Figura 10.1** Gráfico de vazão com e sem o IC.

### *10.2.3 IC para Amostras Pequenas*

Para pequenas amostras, a distribuição Normal apresenta valores menos precisos, o que nos leva a utilizar outro modelo, a distribuição *t* de Student. Quando o tamanho das amostras é grande, por exemplo 100, a distribuição *t* é muito similiar à distribuição Normal. Mas para amostras pequenas, a distribuição *t* é mais precisa.

Para calcular o valor de *t* a ser usado, é necessário estabelecer:

- o nível de confiança desejado;
- o número de graus de liberdade[2] a ser utilizado.

A tabela do Anexo B possui os valores da distribuição *t* a serem utilizados. Usando o símbolo *t* no lugar de *z*, tem-se que o IC, neste caso, é dado por:

$$(\bar{x} \quad t_{[1-\alpha/2;\, n-1]} s/\sqrt{n};\ \bar{x} \quad t_{[1-\alpha/2;\, n-1]} s/\sqrt{n})$$

**Exemplo**

Foram realizadas 25 medidas de vazão de um servidor de banco de dados. A média computada foi de 190 requisições por segundo, com desvio padrão igual a 11. Deseja-se saber, com 95% de certeza, o intervalo de confiança da média.

*Resolução*

Como o número de amostras é muito pequeno (25), será utilizada a tabela *t* para o cálculo do IC.

Tem-se que o número de graus de liberdade é $n - 1 \Rightarrow 25 - 1 = 24$ e o nível de significância é $(1-\alpha) = 0{,}95 \Rightarrow \alpha = 0{,}05$.

Conhecendo o número de graus de liberdade e o nível de confiança desejado, deve-se recorrer à tabela de *t*. O valor desejado deve ser conferido na linha 24 e coluna, $1 - \alpha/2 = 1 - 0{,}05/2 = 0{,}975$, neste caso, igual a 2,064.

Portanto,

$$\text{IC} = (\bar{x} - t_{1-\alpha/2;\, n-1} s/\sqrt{n},\ \bar{x} + t_{1-\alpha/2;\, n-1} s/\sqrt{n})$$

$$\text{IC} = (190 \pm t_{(0{,}975;\, 24)} \times 11/\sqrt{25})$$

$$\text{IC} = (190 \pm 2{,}064 \times 11/\sqrt{25})$$

$$\text{IC} = (185{,}46;\ 194{,}54)$$

O resultado diz que, com 95% de certeza, a média dos valores de tempo de resposta está contida no intervalo de 185,46 a 194,54 milissegundos.

[2] Número de termos independentes em uma somatória.

## 10.3 Comparação de Sistemas

Suponha que quatro sistemas computacionais (A, B, C e D) são comparados em relação a seu tempo de resposta de atender a uma solicitação. A Figura 10.2 mostra o gráfico contendo as médias de tempo de resposta para cada sistema.

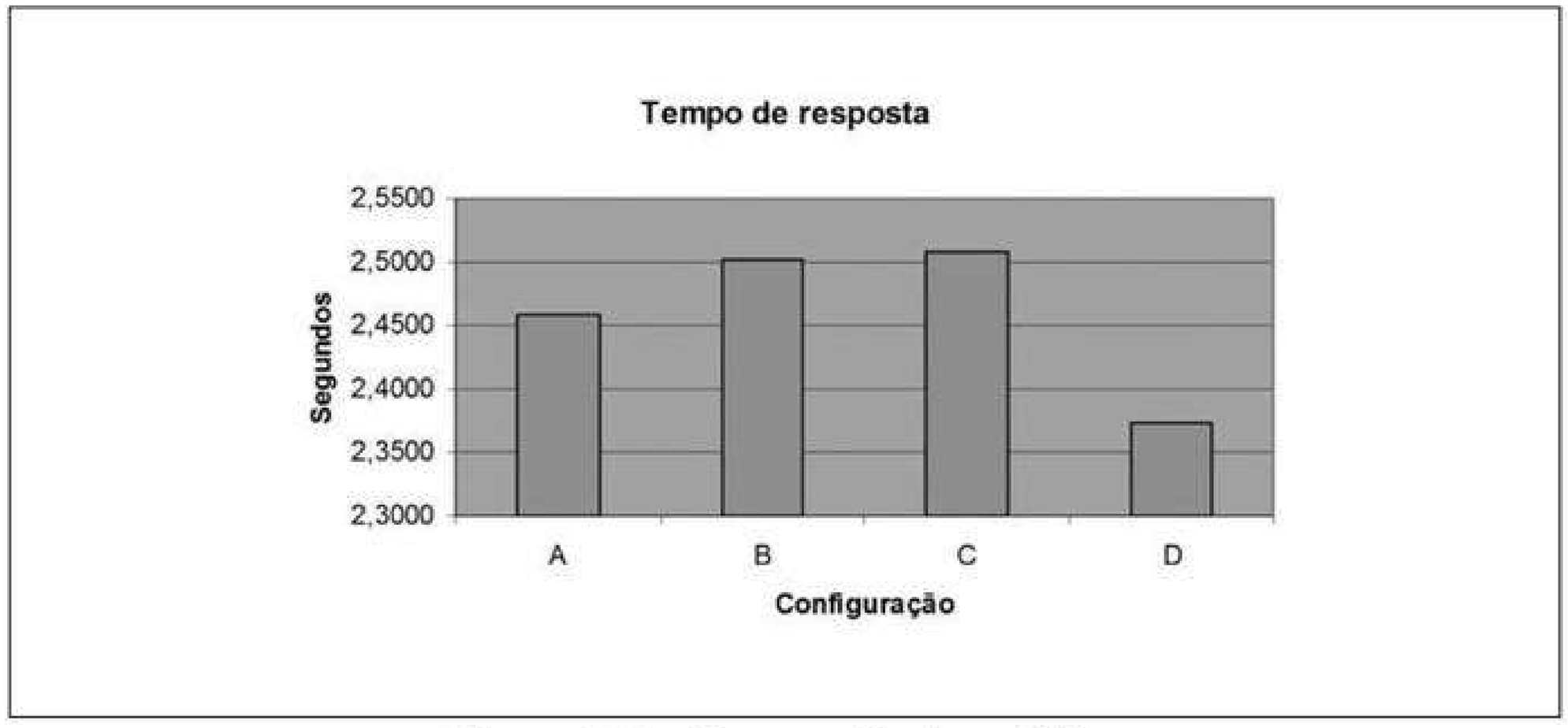


Figura 10.2 Comparação das médias.

Ao analisar o gráfico chega-se à conclusão de que o sistema D possui menor tempo de resposta em relação aos outros sistemas, portanto, seria o sistema com melhor desempenho, certo?

O gráfico anterior é agora refeito, mudando o eixo *y* para acomodar todos os valores entre 0 e 2,55 segundos. O gráfico refeito é mostrado na Figura 10.3.

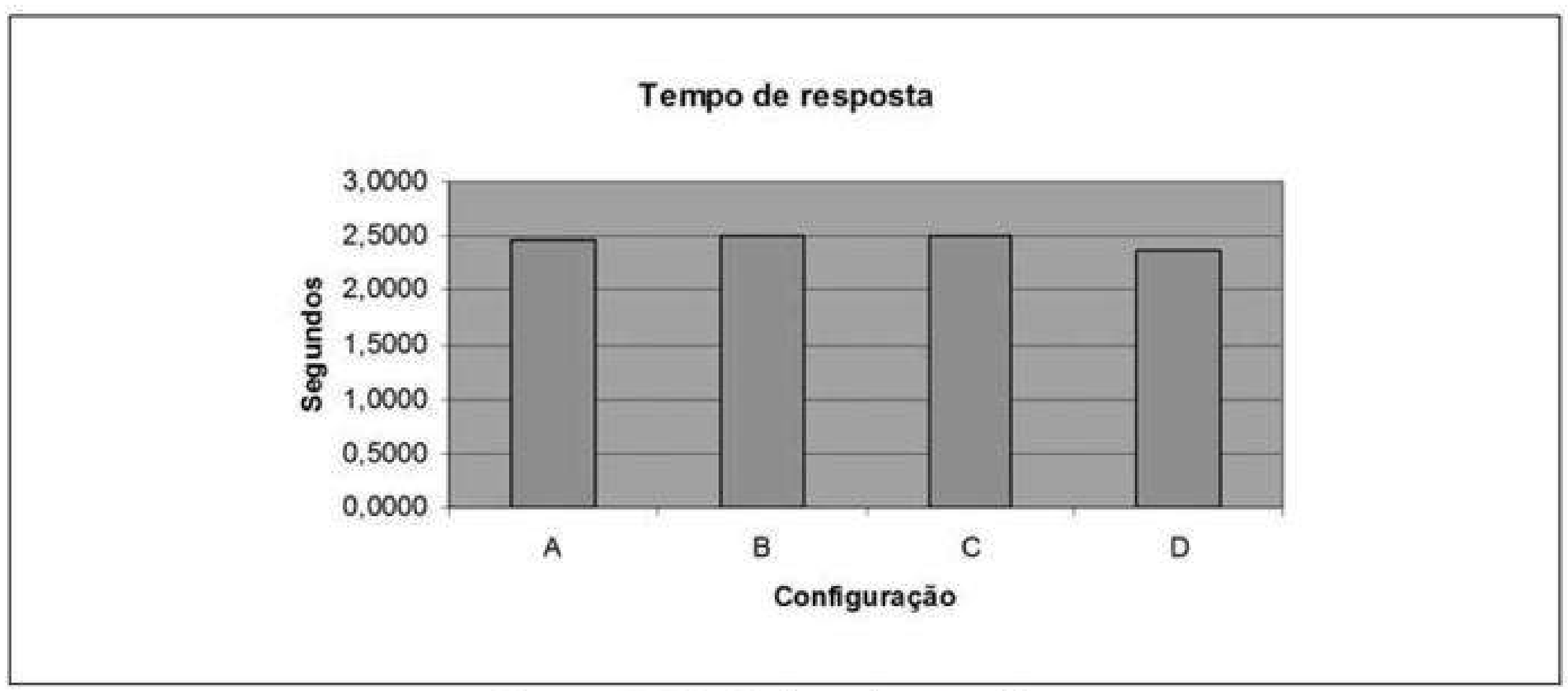


Figura 10.3 Refazendo o gráfico.

Neste novo gráfico, a diferença entre o sistema D e os demais não parece tão significativa quanto no gráfico da Figura 10.2. Mas, aparentemente, o sistema D ainda possui menor tempo de resposta.

O gráfico não mostra os intervalos de confiança da média para cada sistema. A Figura 10.4 mostra o gráfico anterior contendo os ICs para os sistemas.

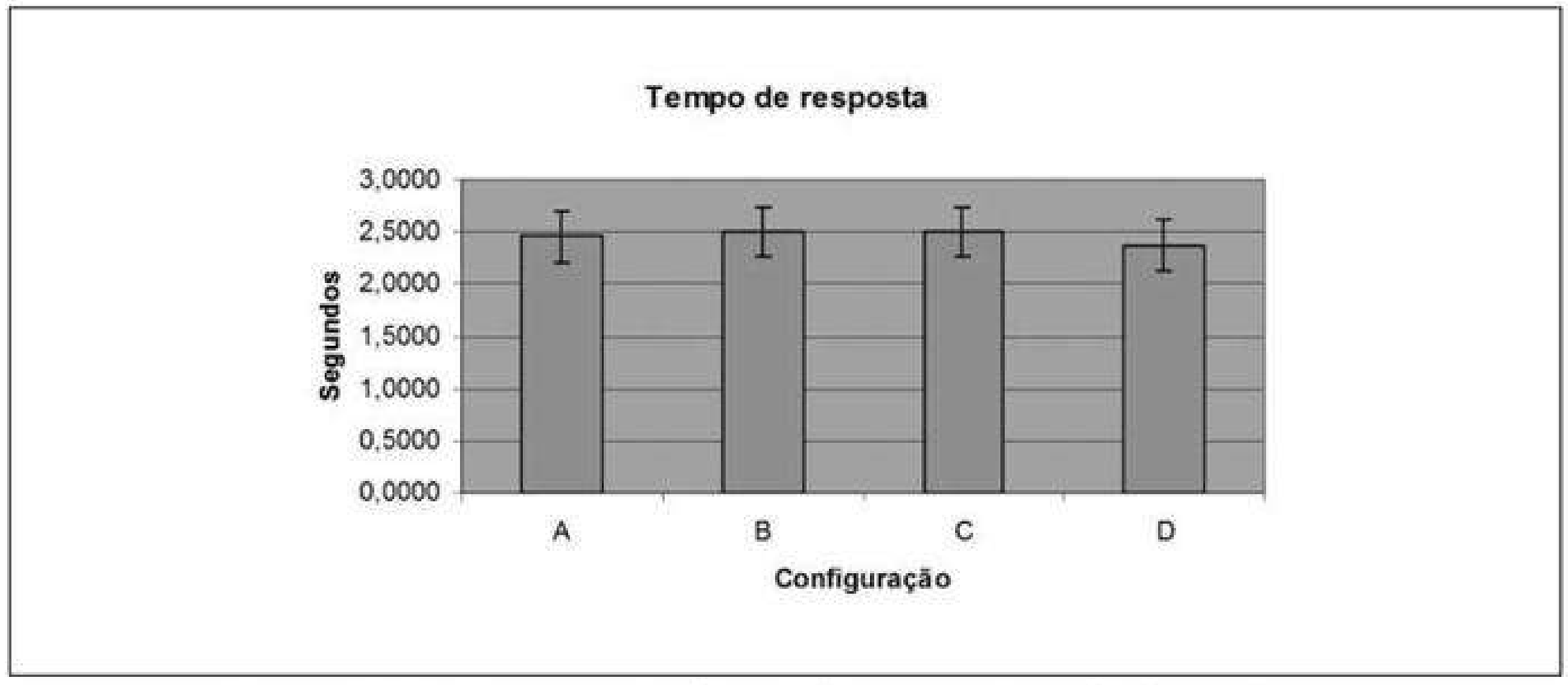


Figura 10.4 Comparação das médias com intervalo de confiança.

O novo gráfico, contendo os ICs, mostra que, na verdade, não é possível afirmar qual sistema possui menor tempo de resposta dentre os quatro.

Esta seção mostrará como fazer a comparação entre os sistemas utilizando quatro técnicas diferentes: o teste da média zero, observações pareadas, observações não pareadas e o teste visual.

### *10.3.1 Teste da Média Zero*

Esse primeiro teste é bem simples e consiste em checar se os valores medidos são significantemente diferentes de zero. Usando o Intervalo de Confiança para o valor medido, verifica-se se o intervalo contém zero.

O uso da média zero para comparar o desempenho de dois sistemas pode ser resumido a:

- utiliza-se a diferença dos valores das amostras de dois sistemas;
- calcula-se o intervalo de confiança da média para o nível de confiança desejado usando os valores da diferença obtidos no passo anterior;
- se o IC resultante incluir a média zero, não é possível dizer que os dois sistemas analisados são diferentes em termos de desempenho; se o IC resultante não incluir a média zero, pode-se afirmar que os sistemas são diferentes, com o nível de confiança desejado.

**Exemplo**

A empresa X analisou dois servidores Firewalls para realizar a compra. Foram feitas oito tentativas de passar um tráfego "suspeito" para dentro da rede da empresa, e computou-se o tempo de resposta para cada tentativa filtrada e rejeitada para saber se os sistemas eram compatíveis em termos de desempenho. Os valores obtidos para a diferença dos tempos de resposta dos dois servidores, em segundos, foram:

$$D = \{1,5; -1,4;\ 0,8; -1,3; -0,5;\ 1,7;\ 0,9;\ 1,1\}$$

Neste caso, é possível dizer, com confiança de 95%, que um servidor é superior ao outro?

*Resolução*

Número de amostras: $n = 8$
Média = 2,8/8 = 0,35
Desvio padrão = 1,24

Com apenas oito amostras, usa-se a distribuição *t*.

Tem-se que o número de graus de liberdade é $n - 1 \Rightarrow 8 - 1 = 7$ e o nível de significância é $(1-\alpha) = 0,95 \Rightarrow \alpha = 0,05$.

$$1-\alpha/2 = 0,975, \text{ então } t_{[0,975;7]} = 2,447$$

Portanto,

$$IC = (0,35 \pm t_{(0,975;7)} \times 1,24/\sqrt{7})$$

$$IC = (0,35 \pm 2,447 \times 1,24/\sqrt{7})$$

$$IC = (-0,79; 1,49)$$

O resultado diz que, com 95% de certeza, a diferença da média dos valores de tempo de resposta está contida no intervalo de −0,79 a 1,49 segundo. Neste caso, não é possível dizer que a diferença é significantemente diferente de zero. Portanto, com 95% de confiança, é possível dizer que o desempenho dos algoritmos é idêntico.

## 10.3.2 Observações Pareadas

Outra forma de comparar dois sistemas é a comparação direta dos valores de IC. Quando houver *n* amostras de um sistema A e *n* amostras de um sistema B, diz-se que existe correspondência 1-para-1.

O uso da média zero para comparar o desempenho de dois sistemas pode ser resumido a:

- trata-se as duas amostras como uma amostra de *n* pares;
- calcula-se a diferença dos valores de cada par de amostras;
- calcula-se o intervalo de confiança da média para o nível de confiança desejado usando os valores da diferença obtidos no passo anterior;
- se o IC resultante incluir a média zero, não é possível dizer que os dois sistemas analisados são diferentes em termos de desempenho; se o IC resultante não incluir a média zero, pode-se afirmar que os sistemas são diferentes, com o nível de confiança desejado.

**Exemplo**

Dois roteadores foram comprados e é necessário fazer uma avaliação em relação à vazão de pacotes por segundo. O conjunto da carga de trabalho dos roteadores A e B está resumido no seguinte conjunto de pares (vazão A, vazão B):

Medidas = {(55,4; 69,1), (80,1; 90,2), (59,1; 55,4), (60,2; 60,1), (62,3; 60,1) ,(55,3; 53,7)}

É possível dizer que um dos roteadores é melhor do que o outro, com confiança de 90%?

*Resolução*

Diferença de medidas = {−13,7; −10,1; 3,7; 0,1; 2,2; 1,6}
Número de amostras: $n = 6$
Média = −16,2/6 = −2,7
Desvio padrão = 7,31

Com apenas seis amostras, usa-se a distribuição *t*.

Tem-se que o número de graus de liberdade é $n - 1 \Rightarrow 6 - 1 = 5$ e o nível de significância é $(1-\alpha) = 0{,}90 \Rightarrow \alpha = 0{,}1$.

$$1-\alpha/2 = 0{,}95$$

então $t_{[0{,}95;5]} = 2{,}015$

Portanto,

$$IC = (-16{,}2 \pm t_{(0{,}95;\,5)} \times (-2{,}7)/\sqrt{6})$$

$$IC = (-16{,}2 \pm 2{,}015 \times (-2{,}7)/\sqrt{6})$$

$$IC = (-18{,}42; -13{,}98)$$

O resultado diz que, com 90% de certeza, a diferença da média dos valores de tempo de resposta está contida no intervalo de −18,42 a −13,98 segundos. Neste caso, é possível dizer que a diferença é significantemente diferente de zero (o intervalo de confiança não contém zero). Portanto, com 90% de confiança, é possível dizer que o desempenho dos Firewalls não é idêntico.

Para saber qual dos dois Firewalls é o melhor em termos de vazão, basta apenas calcular a média da vazão de cada um e verificar qual deles tem a maior vazão, pois já foi possível afirmar que os dois sistemas não são equivalentes.

### *10.3.3 Observações Não Pareadas*

Para comparar, diretamente, dois sistemas não possuem uma correspondência 1-para-1, isto é, um sistema A apresenta *n* amostras, mas o sistema B apresenta *m* amostras, com $n \neq m$, utiliza-se o teste *t*.

## Teste *t*

Têm-se os sistemas A e B, com número de amostras $n_a$ e $n_b$.

- Calculam-se as médias das amostras $\overline{x_a}$ e $\overline{x_b}$
- Calculam-se os desvios padrões $s_a$ e $s_b$
- Calcula-se a diferença da média $\overline{x_a} - \overline{x_b}$

- Calcula-se o desvio padrão da diferença da média $(s)$

$$s = \sqrt{\left(\frac{s_a^2}{n_a} + \frac{s_b^2}{n_b}\right)}$$

- Calculam-se os graus de liberdade

$$v = \frac{\left(\frac{s_a^2}{n_a} + \frac{s_b^2}{n_b}\right)^2}{\frac{1}{n_a+1}\left(\frac{s_a^2}{n_a}\right)^2 + \frac{1}{n_b+1}\left(\frac{s_b^2}{n_b}\right)^2} - 2$$

- Calcula-se o intervalo de confiança da diferença da média

$$(\overline{x}_a - \overline{x}_b) \pm t_{[1-\alpha/2;\, v]} s$$

Se o Intervalo de Confiança resultante incluir zero, então os sistemas comparados não possuem uma diferença significante um do outro.

**Exemplo**

Dois sistemas de processamento de requisições Web foram medidos em relação ao tempo de atendimento de requisições. Os valores medidos, em segundos, para os sistemas foram:

A = {1,22; 1,23; 1,67; 0,98; 0,88; 3,34}
B = {0,99; 1,48; 3,88; 1,20; 2,09}

Deseja-se saber, com 90% de certeza, se os dois sistemas são significantemente diferentes.

*Resolução*

Como os dois sistemas apresentam número diferente de amostras ($n_a = 6$ e $n_b = 5$), deve-se usar a comparação de sistemas com observações não-pareadas. Segue-se então o teste *t*.

Média $\overline{x_a} = 1{,}55$

*(continua)*

Média $\overline{x_b} = 1{,}93$

Diferença da média $\overline{x_a} - \overline{x_b} = -0{,}38$

Desvio padrão $s_a = 0{,}92$

Desvio padrão $s_b = 1{,}17$

Desvio padrão da diferença da média

$$s = \sqrt{\left(\frac{s_a^2}{n_a} + \frac{s_b^2}{n_b}\right)} = \sqrt{\left(\frac{0{,}92^2}{6} + \frac{1{,}17^2}{5}\right)} = 0{,}64$$

Graus de liberdade

$$v = \frac{\left(\frac{s_a^2}{n_a} + \frac{s_b^2}{n_b}\right)^2}{\frac{1}{n_a+1}\left(\frac{s_a^2}{n_a}\right)^2 + \frac{1}{n_b+1}\left(\frac{s_b^2}{n_b}\right)^2} - 2$$

$$v = \frac{\left(\frac{0{,}92^2}{6} + \frac{1{,}17^2}{5}\right)^2}{\frac{1}{6+1}\left(\frac{0{,}92^2}{6}\right)^2 + \frac{1}{5+1}\left(\frac{1{,}17^2}{5}\right)^2} - 2 = \frac{0{,}17}{0{,}0028 + 0{,}012} - 2 = 9{,}49$$

Deve-se arredondar o valor de $v$ para o inteiro mais próximo. Portanto, $v = 9{,}0$. O nível de significância é $(1-\alpha) = 0{,}90 \Rightarrow \alpha = 0{,}1$

$$1 - \alpha/2 = 0{,}95$$

então

$$t_{[0{,}95;\,9]} = 1{,}833$$

Portanto, o intervalo de confiança da diferença da média

$$\text{IC} = (\overline{x_a} - \overline{x_b}) \pm t_{[1-\alpha/2;\,v]}s$$

$$\text{IC} = (-0{,}38) \pm t_{[0{,}95;\,9]}0{,}64$$

$$\text{IC} = (-0{,}38) \pm 1{,}833 \times 0{,}64$$

$$\text{IC} = (-1{,}55;\ 0{,}79)$$

Como o intervalo de confiança contém zero, pode-se afirmar, com 90% de certeza, que os sistemas são idênticos.

### 10.3.4 Teste Visual

No teste visual, utiliza-se a comparação de sistemas através da sobreposição dos intervalos de confiança para os sistemas analisados. São possíveis três opções:

- **Intervalos de confiança não se sobrepõem**

Nesse primeiro caso, os intervalos de confiança e as médias dos sistemas avaliados não se sobrepõem.

Por exemplo, os sistemas A e B foram analisados quanto à vazão, e calculou-se os ICs para cada um: $IC_A = (12{,}4;\ 18{,}8)$ e $IC_B = (19{,}3;\ 24{,}9)$. Claramente, os intervalos não se sobrepõem.

Quando a não sobreposição ocorre, a análise comparativa entre os dois sistemas pode se basear diretamente na avaliação da métrica usada. Por exemplo, ao avaliar dois sistemas em relação à vazão, verifica-se qual obteve maior/menor vazão utilizando os valores médios.

- **Intervalos de confiança e médias se sobrepõem**

Quando os intervalos de confiança e as médias dos sistemas avaliados se sobrepõem, não é possível dizer que os dois sistemas são diferentes. Então, a avaliação das métricas utilizadas para cálculo do IC só pode afirmar que os sistemas são idênticos.

Por exemplo, os sistemas A e B foram observados e os seguintes dados de tempo de resposta foram coletados e analisados:

$$\text{Média}_A = 14{,}4;\ IC_A = (12{,}4;\ 18{,}8)$$
$$\text{Média}_B = 16{,}1;\ IC_B = (13{,}0;\ 17{,}9)$$

Vê-se que a média de A está contida no intervalo de B, e a média de B está contida no intervalo de A. Portanto, não é possível dizer que os sistemas são diferentes.

- **Intervalos de confiança se sobrepõem, mas as médias não**

Quando os ICs dos sistemas avaliados se sobrepõem, mas a média de um sistema não sobrepõe no IC do outro sistema, não é possível ter certeza que os sistemas são ou não idênticos.

Como exemplo, os sistemas A e B possuem os seguintes dados de tempo de resposta:

$$\text{Média}_A = 14{,}4;\ IC_A = (12{,}4;\ 18{,}2)$$
$$\text{Média}_B = 18{,}4;\ IC_B = (13{,}0;\ 24{,}9)$$

Verifica-se que a média de A está contida no intervalo de B, mas a média de B não está contida no intervalo de A.

Para resolver esta dúvida, realiza-se o teste *t* (Subseção 10.3.3).

**Exemplo 1**

Os tempos de resposta de dois servidores para uma mesma tarefa foram medidos e os ICs com 90% foram calculados em:

$$\text{Média}_A = 10{,}71\ ,\ IC_A = (10{,}24;\ 10{,}98)$$

$$\text{Média}_B = 10{,}91,\ IC_B = (10{,}18;\ 12{,}23)$$

Os dois intervalos de confiança se sobrepõem e a média do sistema A se encontra no intervalo do sistema B, e vice-versa. Portanto, com 90% de certeza, podemos afirmar que os dois sistemas não são diferentes.

**Exemplo 2**

O tempo de resposta do processador gasto na execução de uma determinada tarefa foi medido em dois servidores, A e B. Os tempos de resposta dos sistemas, em milissegundos, foram:

$$R_A = \{0{,}33;\ 0{,}46;\ 1{,}32;\ 1{,}44;\ 0{,}57;\ 1{,}12\}$$

$$R_B = \{1{,}2;\ 2{,}44;\ 1{,}01;\ 1{,}45;\ 2{,}89;\ 1{,}88\}$$

Quer-se saber se os sistemas são significantemente diferentes, com 95% de certeza.

*Resolução*

Média $\overline{x_a} = 0{,}87$

Média $\overline{x_b} = 1{,}80$

Número de amostras $n_a = n_b = 6$
Desvio padrão $s_a = 0{,}48$
Desvio padrão $s_b = 0{,}74$

Diferença média $\overline{x_a} - \overline{x_b} = -0{,}93$

Desvio padrão da diferença média, $s$

$$s = \sqrt{\left(\frac{s_a^2}{n_a} + \frac{s_b^2}{n_b}\right)} = \sqrt{\left(\frac{0{,}48^2}{6} + \frac{1{,}8^2}{6}\right)} = 0{,}76$$

Número efetivo de graus de liberdade

$$\nu = \frac{\left(\frac{0{,}87^2}{6} + \frac{1{,}80^2}{6}\right)^2}{\frac{1}{6+1}\left(\frac{0{,}87^2}{6}\right)^2 + \frac{1}{6+1}\left(\frac{1{,}80^2}{6}\right)^2} - 2 = \frac{0{,}58}{0{,}0023 + 0{,}042} - 2 = 11{,}09 \approx 11$$

*(continua)*

O nível de significância é $(1-\alpha) = 0{,}95 \Rightarrow \alpha = 0{,}05$

$$1-\alpha/2 = 0{,}975 \text{ então } t_{[0{,}975;\,11]} = 2{,}201$$

Portanto, calculando-se o intervalo de confiança da diferença da média, tem-se que

$$\text{IC} = (\bar{x}_a - \bar{x}_b) \pm t_{[1-\alpha/2;\,\nu]} s$$

$$\text{IC} = (-0{,}93) \pm t_{[0{,}975;\,11]} 0{,}76$$

$$\text{IC} = (-0{,}93) \pm 2{,}201 \times 0{,}76$$

$$\text{IC} = (-2{,}60;\ 0{,}74)$$

Como o intervalo de confiança contém zero, pode-se afirmar, com 95% de certeza, que os sistemas são idênticos.

## 10.4 Exercícios

1) Uma empresa comprou uma nova máquina para ser servidora de bancos de dados. Diversos testes de vazão (transações por segundo) foram realizados e o seguinte conjunto de dados foi obtido:
Vazão = {43,1; 59,2; 55,2; 44,3; 40,33; 37,0; 80,1; 60,2; 61,3; 120,1; 41,2; 50,2; 50,2; 46,3; 47,2; 57,0; 44,1; 61,2; 63,3; 30,1; 41,3; 56,6; 45,2; 64,3; 60,3; 77,0; 40,1; 50,2; 71,3; 50,1; 47,1; 55,5; 58,1; 40,0; 41,9; 47,0; 39,1; 40,2; 54,3; 61,1}

   Deseja-se saber, com 90% de confiança, a vazão média do sistema.

2) Foram realizados 50 experimentos de simulação para análise de um novo protocolo para transmissão de dados em redes de computadores. Foram executados 25 experimentos para o protocolo A, padrão de indústria, e o restante foram experimentos do novo protocolo Y. Os seguintes dados foram coletados, e precisa-se saber, com 95% de certeza, que os protocolos são diferentes.

| Protocolo | Média do tempo de resposta (s) | Desvio padrão |
|---|---|---|
| A | 1,45 | 0,34 |
| Y | 1,20 | 0,23 |

3) Um servidor Web foi testado para verificar se mudanças em sua configuração afetaram o desempenho do sistema. Os seguintes conjuntos de dados para tempo de resposta, em milissegundos, foram obtidos:

$R_{antes} = \{3,1; 9,2; 5,2; 4,3; 3,33; 6,0; 7,1; 4,2\}$
$R_{depois} = \{2,1; 2,2; 3,2; 3,3; 2,33; 6,0; 1,1; 1,9\}$

Deseja-se saber, com 99% de confiança, se as modificações surtiram efeito.

4) Os seguintes conjuntos de dados para tempo de resposta, em milissegundos, foram obtidos para dois computadores diferentes processando a mesma transação:

$R_{computador1} = \{10,2; 12,3; 13,4; 14,0; 17,1; 10,2\}$
$R_{computador2} = \{12,2; 13,2; 13,3; 12,0; 10,0; 11,1; 16,9\}$

Deseja-se saber, com 95% de confiança, qual o computador mais rápido.

# Anexos

## ANEXO A Quantil de uma Distribuição Normal Padronizada

| P | 0,00 | 0,01 | 0,02 | 0,03 | 0,04 | 0,05 | 0,06 | 0,07 | 0,08 | 0,09 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,5 | 0,000 | 0,025 | 0,050 | 0,075 | 0,010 | 0,126 | 0,151 | 0,176 | 0,202 | 0,228 |
| 0,6 | 0,253 | 0,279 | 0,305 | 0,332 | 0,358 | 0,385 | 0,412 | 0,440 | 0,468 | 0,496 |
| 0,7 | 0,524 | 0,553 | 0,583 | 0,613 | 0,643 | 0,674 | 0,706 | 0,739 | 0,772 | 0,806 |
| 0,8 | 0,842 | 0,878 | 0,915 | 0,954 | 0,994 | 1,036 | 1,080 | 1,126 | 1,175 | 1,227 |
| | | | | | | | | | | |
| P | 0,000 | 0,001 | 0,002 | 0,003 | 0,004 | 0,005 | 0,006 | 0,007 | 0,008 | 0,009 |
| 0,90 | 1,282 | 1,287 | 1,293 | 1,299 | 1,305 | 1,311 | 1,317 | 1,323 | 1,329 | 1,335 |
| 0,91 | 1,341 | 1,347 | 1,353 | 1,359 | 1,366 | 1,372 | 1,379 | 1,385 | 1,392 | 1,398 |
| 0,92 | 1,405 | 1,412 | 1,419 | 1,426 | 1,433 | 1,440 | 1,447 | 1,454 | 1,461 | 1,468 |
| 0,93 | 1,476 | 1,483 | 1,491 | 1,499 | 1,506 | 1,514 | 1,522 | 1,530 | 1,538 | 1,546 |
| 0,94 | 1,555 | 1,563 | 1,572 | 1,580 | 1,589 | 1,598 | 1,607 | 1,616 | 1,626 | 1,635 |
| 0,95 | 1,645 | 1,655 | 1,665 | 1,675 | 1,685 | 1,695 | 1,706 | 1,717 | 1,728 | 1,739 |
| 0,96 | 1,751 | 1,762 | 1,774 | 1,787 | 1,799 | 1,812 | 1,825 | 1,838 | 1,852 | 1,866 |
| 0,97 | 1,881 | 1,896 | 1,911 | 1,927 | 1,943 | 1,960 | 1,977 | 1,995 | 2,014 | 2,034 |
| 0,98 | 2,054 | 2,075 | 2,097 | 2,120 | 2,144 | 2,170 | 2,197 | 2,226 | 2,257 | 2,290 |
| | | | | | | | | | | |
| P | 0,0000 | 0,0001 | 0,0002 | 0,0003 | 0,0004 | 0,0005 | 0,0006 | 0,0007 | 0,0008 | 0,0009 |
| 0,990 | 2,326 | 2,330 | 2,334 | 2,338 | 2,342 | 2,346 | 2,349 | 2,353 | 2,357 | 2,362 |
| 0,991 | 2,366 | 2,370 | 2,374 | 2,378 | 2,382 | 2,387 | 2,391 | 2,395 | 2,400 | 2,404 |
| 0,992 | 2,409 | 2,414 | 2,418 | 2,423 | 2,428 | 2,432 | 2,437 | 2,442 | 2,447 | 2,452 |
| 0,993 | 2,457 | 2,462 | 2,468 | 2,473 | 2,478 | 2,484 | 2,489 | 2,495 | 2,501 | 2,506 |
| 0,994 | 2,512 | 2,518 | 2,524 | 2,530 | 2,536 | 2,543 | 2,549 | 2,556 | 2,562 | 2,569 |
| 0,995 | 2,576 | 2,583 | 2,590 | 2,597 | 2,605 | 2,612 | 2,620 | 2,628 | 2,636 | 2,644 |
| 0,996 | 2,652 | 2,661 | 2,669 | 2,678 | 2,687 | 2,697 | 2,706 | 2,716 | 2,272 | 2,737 |
| 0,997 | 2,748 | 2,759 | 2,770 | 2,782 | 2,794 | 2,807 | 2,820 | 2,824 | 2,848 | 2,863 |
| 0,998 | 2,878 | 2,894 | 2,911 | 2,929 | 2,948 | 2,968 | 2,989 | 3,011 | 3,036 | 3,062 |
| 0,999 | 3,090 | 3,121 | 3,156 | 3,195 | 3,239 | 3,291 | 3,353 | 3,432 | 3,540 | 3,719 |

## ANEXO B Quantil de uma Distribuição *t*

| n | 0,6000 | 0,7000 | 0,8000 | 0,9000 | 0,9500 | 0,9750 | 0,9950 | 0,9995 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0,325 | 0,727 | 1,377 | 3,078 | 6,314 | 12,706 | 63,657 | 636,619 |
| 2 | 0,289 | 0,617 | 1,061 | 1,886 | 2,920 | 4,303 | 9,925 | 31,599 |
| 3 | 0,277 | 0,584 | 0,978 | 1,638 | 2,353 | 3,182 | 5,841 | 12,924 |
| 4 | 0,271 | 0,569 | 0,941 | 1,533 | 2,132 | 2,776 | 4,604 | 8,610 |
| 5 | 0,267 | 0,559 | 0,920 | 1,476 | 2,015 | 2,571 | 4,032 | 6,869 |
| 6 | 0,265 | 0,553 | 0,906 | 1,440 | 1,943 | 2,447 | 3,707 | 5,959 |
| 7 | 0,263 | 0,549 | 0,896 | 1,415 | 1,895 | 2,365 | 3,499 | 5,408 |
| 8 | 0,262 | 0,546 | 0,889 | 1,397 | 1,860 | 2,306 | 3,355 | 5,041 |
| 9 | 0,261 | 0,543 | 0,883 | 1,383 | 1,833 | 2,262 | 3,250 | 4,781 |
| 10 | 0,260 | 0,542 | 0,879 | 1,372 | 1,812 | 2,228 | 3,169 | 4,587 |
| 11 | 0,260 | 0,540 | 0,876 | 1,363 | 1,796 | 2,201 | 3,106 | 4,437 |
| 12 | 0,259 | 0,539 | 0,873 | 1,356 | 1,782 | 2,179 | 3,055 | 4,318 |
| 13 | 0,259 | 0,538 | 0,870 | 1,350 | 1,771 | 2,160 | 3,012 | 4,221 |
| 14 | 0,258 | 0,537 | 0,868 | 2,345 | 1,761 | 2,145 | 2,977 | 4,140 |
| 15 | 0,258 | 0,536 | 0,866 | 1,341 | 1,753 | 2,131 | 2,947 | 4,073 |
| 16 | 0,258 | 0,535 | 0,865 | 1,337 | 1,746 | 2,120 | 2,921 | 4,015 |
| 17 | 0,257 | 0,534 | 0,863 | 1,333 | 1,740 | 2,110 | 2,898 | 3,965 |
| 18 | 0,257 | 0,534 | 0,862 | 1,330 | 1,734 | 2,101 | 2,878 | 3,922 |
| 19 | 0,257 | 0,533 | 0,861 | 1,328 | 1,729 | 2,093 | 2,861 | 3,883 |
| 20 | 0,257 | 0,533 | 0,860 | 1,325 | 1,725 | 2,086 | 2,845 | 3,850 |
| 21 | 0,257 | 0,532 | 0,859 | 1,323 | 1,721 | 2,080 | 2,831 | 3,819 |
| 22 | 0,256 | 0,532 | 0,858 | 1,321 | 1,717 | 2,074 | 2,819 | 3,792 |
| 23 | 0,256 | 0,532 | 0,858 | 1,319 | 1,714 | 2,069 | 2,807 | 3,768 |
| 24 | 0,256 | 0,531 | 0,857 | 1,318 | 1,711 | 2,064 | 2,797 | 3,745 |
| 25 | 0,256 | 0,531 | 0,856 | 1,316 | 1,708 | 2,060 | 2,787 | 3,725 |
| 26 | 0,256 | 0,531 | 0,856 | 1,315 | 1,706 | 2,056 | 2,779 | 3,707 |
| 27 | 0,256 | 0,531 | 0,855 | 1,314 | 1,703 | 2,052 | 2,771 | 3,690 |
| 28 | 0,256 | 0,530 | 0,855 | 1,313 | 1,701 | 2,048 | 2,763 | 3,674 |
| 29 | 0,256 | 0,530 | 0,854 | 1,311 | 1,699 | 2,045 | 2,756 | 3,659 |
| 30 | 0,256 | 0,530 | 0,854 | 1,310 | 1,697 | 2,042 | 2,750 | 6,646 |
| 60 | 0,254 | 0,527 | 0,848 | 1,296 | 1,671 | 2,000 | 2,660 | 3,460 |
| 90 | 0,254 | 0,526 | 0,846 | 1,291 | 1,662 | 1,987 | 2,632 | 3,402 |
| 120 | 0,254 | 0,536 | 0,845 | 1,289 | 1,658 | 1,980 | 2,617 | 3,373 |

# Estudo de Caso: Modelagem de Sistemas Web

# 11

## 11.1 Introdução

A universidade XYZ possui um sistema de educação remota via Internet que é muito popular. Os estudantes podem assistir às aulas a qualquer hora do dia e da noite, usando o *browser* de seus computadores para acessar o material, que consiste em aulas de vídeo, áudio e texto. Em certas horas do dia o sistema está sobrecarregado, pois muitos alunos estão acessando o sistema ao mesmo tempo.

O sistema Web de aulas deve ser analisado para verificar qual ou quais são os gargalos que afetam o desempenho do servidor.

A arquitetura do sistema Web pode ser vista na Figura 11.1.

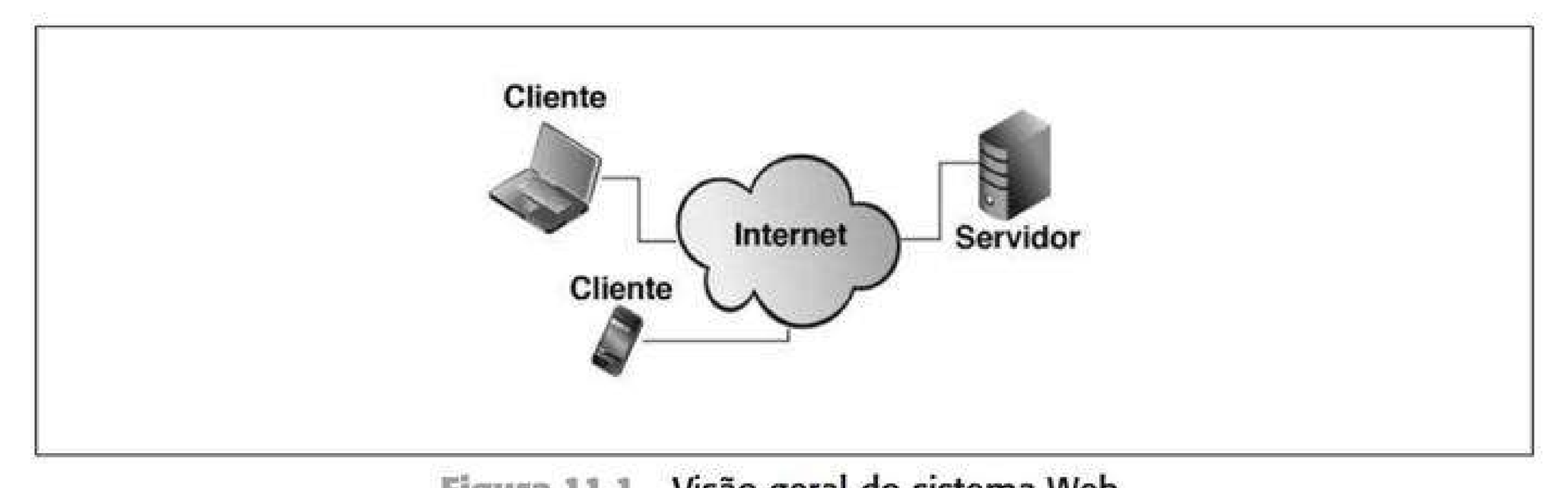


**Figura 11.1** Visão geral do sistema Web.

Ele consiste em um servidor *Apache*, que, usando mensagens de protocolo HTTP, pode trocar informações com o software do aluno, neste caso, um *browser*. O computador do aluno pode ser um *desktop*, um *laptop*, um *tablet* (*iPad* e afins), *smartphone* etc., que possua um *browser Web*.

O aluno pode baixar os arquivos de aulas e enviar respostas de exercícios, dúvidas etc. através deste sistema. Para baixar arquivos, basta o aluno clicar no item desejado e aguardar o download. Para interação com o professor ou o sistema acadêmico, existem páginas para o aluno preencher formulários e enviar os dados.

A análise deste capítulo será direcionada para o tráfego de arquivos e o processamento entre o servidor e o cliente.

## 11.2 Transação Web

Uma transação Web corresponde a vários passos:

1) O cliente realiza um pedido de processamento. Por exemplo, no *browser*, o cliente clica no nome do arquivo desejado.
2) O *browser* usa o protocolo HTTP (*Hypertext Transfer Protocol*) para a transferência de informações a um servidor remoto.
3) O "pacote" da requisição gerado pelo protocolo HTTP[1] é recebido no servidor Web para processamento.
4) O servidor Web demora um certo tempo $\Delta t$ para realizar o processamento da transação recebida.
5) Utilizando o protocolo HTTP, o servidor envia a resposta do processamento ao cliente.
6) O *browser/software* do cliente recebe a resposta e mostra o resultado ao cliente.

A Figura 11.2 exemplifica o processo da transação Web.

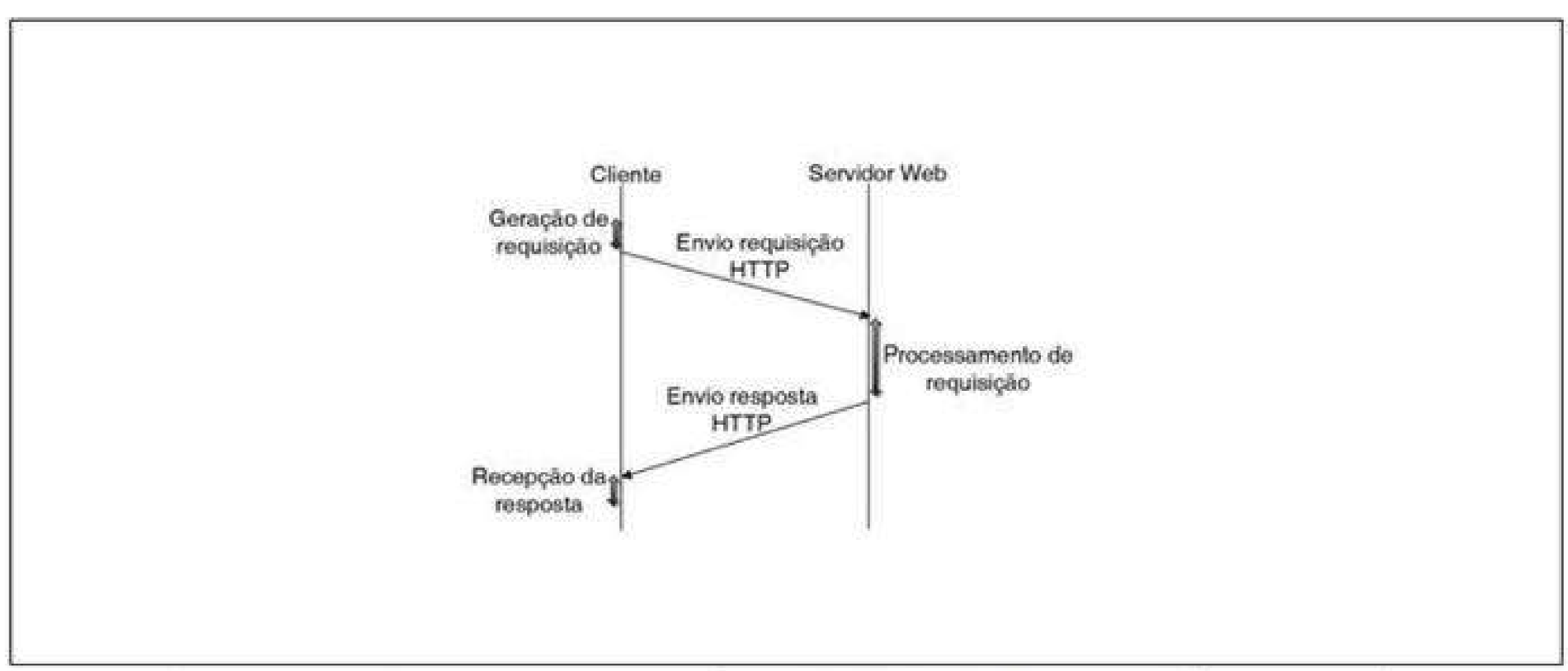


**Figura 11.2** Transação através de requisição HTTP processada no servidor.

No restante deste capítulo os lados Cliente e Servidor da transação Web serão vistos e analisados em relação ao desempenho das principais métricas, como vazão e tempo de resposta.

## 11.3 Desempenho no Lado Servidor

O servidor Web estudado processa dois tipos de carga de trabalho: páginas estáticas e páginas dinâmicas. Como o servidor em análise é usado somente para busca de conteúdo de aulas, supõe-se que o conteúdo é disponibilizado em páginas HTML estáticas.

Para páginas estáticas, o servidor, após receber a requisição do aluno, deve:

[1] Neste exemplo, o modelo de camadas TCP/IP será omitido.

a) ler o conteúdo do disco;
b) processar a resposta e
c) enviar a resposta.

### *Demanda do Servidor*

A demanda total no servidor Web é composta por todas as demandas nos dispositivos internos utilizados para receber, processar e devolver a transação:

$$D_{\text{servidor}} = j_{\text{visitas}} D_{ij}$$

No caso estudado, será levado em consideração o tempo de processamento da recepção/envio da transação e tempo de acesso a disco para busca dos arquivos.

$$D_{\text{servidor}} = D_{\text{CPU,recepção}} + D_{\text{E/S}} + D_{\text{CPU,envio}}$$

### *Demanda de E/S*

A demanda no disco do servidor é composta pelo tempo de buscar os dados em disco, pela latência de procura e pelo tempo para transferir os arquivos do disco para a memória. O tempo de serviço do disco é:

$$S_{\text{disco}} = \text{TempoBusca} + \text{LatênciaBusca} + \text{TempoTransferência}$$

sendo o *Tempo de Transferência* dado por:

$$\textbf{TempoTransferência} = \textbf{TamanhoBloco} \times \textbf{TaxaTransferência}$$

**Exemplo**

Os tipos de arquivos disponibilizados no servidor e os respectivos tamanhos estão listados na Tabela 11.1.

**TABELA 11.1** Conteúdos e tamanhos

| Conteúdo | Tamanho (Mbytes) |
|---|---|
| Vídeo aula 1 | 45 |
| Vídeo aula 2 | 20 |
| Áudio aula 3 | 8 |
| Áudio aula 4 | 5 |
| Texto aula 4 | 2 |

Calcula-se, então, a demanda do servidor para cada tipo de conteúdo.

O disco do servidor possui bloco de dados de tamanho igual a 2048 bytes e taxa de transferência de 3 Gbytes por segundo. O tempo de busca do disco é de 2 ms e a latência de busca é de 2,2 ms.

*(continua)*

Portanto, o tempo de transferência para 1 bloco é dado por:

$$\textbf{TempoTransferência} = \frac{\textbf{2.048}}{\textbf{3.000.000.000}} = \textbf{6,83} \times \textbf{10}^{-7}\ \textbf{segundos}$$

O tempo de serviço do disco, para 1 bloco de arquivo, é então:

$$S_{disco} = 0{,}02 + 0{,}0022 + 6{,}83 \times 10^{-7} = 0{,}00420068 \text{ segundo}$$

A demanda de serviço do disco é dada pelo número de visitas (blocos a serem lidos) vezes o tempo de serviço do disco (tempo de transferência mais os valores de tempo de busca e latência de busca):

$$D_{E/S} = \text{NumBloco} \times S_{disco}$$

A Tabela 11.2 mostra o número de blocos lidos para cada arquivo e a demanda de serviço do disco para cada tipo de arquivo.

**TABELA 11.2** Demanda do disco do servidor

| Conteúdo | Número de blocos | $D_{disco}$ (segundos) |
|---|---|---|
| Vídeo aula 1 | $45 \times 10^6 / 2048 = 21972{,}65$ | 92,30 |
| Vídeo aula 2 | $20 \times 10^6 / 2048 = 9765{,}63$ | 41,02 |
| Áudio aula 3 | $8 \times 10^6 / 2048 = 3906{,}25$ | 16,41 |
| Áudio aula 4 | $5 \times 10^6 / 2048 = 2441{,}41$ | 10,25 |
| Texto aula 4 | $2 \times 10^6 / 2048 = 976{,}56$ | 4,10 |

Neste exemplo supõe-se que o tempo de recepção e geração de pacotes seja desprezível.[2] Portanto, a demanda do servidor será equivalente à demanda do sistema de E/S.

## 11.4 Desempenho da Transmissão dos Arquivos

O tempo de transmissão de um arquivo na rede de comunicação depende da tecnologia usada para a transmissão. A Tabela 11.3 mostra alguns valores de taxa de transmissão em tecnologias sem fio e cabeadas populares.

**TABELA 11.3** Velocidade dos links de comunicação

| Tipo de rede | Largura de banda (vazão) em MBps |
|---|---|
| Wi-Fi 802.11b | 11 |
| Wi-Fi 802.11g | 54 |
| Cabeada Fast Ethernet | 100 |
| Bluetooth | 3 |
| 3G | 7,2 |

[2] Como os detalhes de TCP/IP foram omitidos, no exemplo são considerados desprezíveis.

O tempo de serviço da rede de comunicação, ou tempo necessário para transferir uma mensagem (textual, arquivos etc.)[3], é dado por:

$$S_{COM} = \frac{8 \times TamanhoMensagem}{10^6 \times LarguraBanda}$$

O tamanho da mensagem é convertido em bits, pois a vazão ou velocidade de uma rede de transmissão de dados possui unidade em bits por segundo.

**Exemplo**

A Tabela 11.4 mostra a configuração das placas de comunicação dos computadores de 4 alunos.

Como o modelo Web é composto de uma requisição e uma resposta, serão calculadas as duas partes da transmissão de dados. Esse cálculo pode também ser feito de uma só vez, somando o tamanho total de dados a serem transferidos, para uso em uma análise fim a fim pelo lado do usuário. Porém, para verificar a diferença entre os tempos de transmissão com uso de diferentes tecnologias, neste exemplo o cálculo é feito em separado.

Para cada aluno, foi calculada a velocidade de transmissão de uma requisição de arquivo. O tamanho da requisição é estimado em 1 Kbyte.

**TABELA 11.4** Velocidade da transmissão por tipo de arquivo

| Aluno | Mensagem | Tipo de rede | Tempo de transmissão (segundos) |
|---|---|---|---|
| Paulo | 1 KB | Wi-Fi 802.11g | $S_{COM} = \frac{8 \times 1}{10^6 \times 54} = 1{,}48e\text{-}7$ |
| Evandro | 1 KB | Wi-Fi 802.11g | $S_{COM} = \frac{8 \times 1}{10^6 \times 11} = 7{,}278e\text{-}7$ |
| Fábio | 1 KB | Fast Ethernet | $S_{COM} = \frac{8 \times 1}{10^6 \times 100} = 8e\text{-}8$ |
| Tânia | 1 KB | 3G | $S_{COM} = \frac{8 \times 1}{10^6 \times 7{,}2} = 1{,}11e\text{-}6$ |

A Tabela 11.5 mostra o tempo de transmissão, para cada aluno, do arquivo desejado.

**TABELA 11.5** Velocidade da transmissão por tipo de arquivo

| Aluno | Mensagem | Tipo de rede | Tempo de transmissão (segundos) |
|---|---|---|---|
| Paulo | Vídeo aula 1 | Wi-Fi 802.11g | $S_{COM_2} = \frac{8 \times 45 \times 1024 \times 1024}{10^6 \times 54} = 7$ |
| Evandro | Vídeo aula 2 | Wi-Fi 802.11b | $S_{COM_2} = \frac{8 \times 20 \times 1024 \times 1024}{10^6 \times 11} = 167{,}77$ |
| Fábio | Vídeo aula 3 | Fast Ethernet | $S_{COM_2} = \frac{8 \times 8 \times 1024 \times 1024}{10^6 \times 100} = 0{,}67$ |
| Tânia | Áudio e texto aula 4 | 3G | $S_{COM_2} = \frac{8 \times (5+2) \times 1024 \times 1024}{10^6 \times 7{,}2} = 8{,}16$ |

[3] Sem incluir a questão de fragmentação dos dados em múltiplos pacotes e camadas, considerando que o arquivo já está na fila de saída.

## 11.5 Desempenho Percebido pelo Cliente

No lado do cliente, deve-se analisar como a transação é percebida pelo cliente. A anatomia da transação é composta, basicamente, de:

- cliente clica em um *hyperlink*;
- procura no *cache* local;
- se não acha, abre conexão e envia requisição;
- recebe e processa a resposta.

O desempenho visto pelo cliente pode ser resumido pelo tempo de resposta total entre o pedido da requisição, sua transmissão na rede, processamento pelo servidor e transmissão de volta:

$$R_{\text{sistema}} = \sum_{\text{recursos}} R_i$$

$$R_{\text{sistema}} = R_{\text{servidor}} + R_{\text{COM}} + R_{\text{cliente}}$$

**Exemplo**

Este exemplo mostra como é percebido o desempenho da transação pelo cliente, considerando que o tempo de processamento local é desprezível e que não existe busca ao *cache* local.

Os dois exemplos anteriores já calcularam os tempos de processamento no servidor dos arquivos e tempo de transmissão para cada usuário. Agora, deseja-se verificar o tempo total percebido por cada usuário.

A Tabela 11.6 mostra os tempos totais percebidos por cada usuário.

**TABELA 11.6** Velocidade da transmissão por tipo de arquivo

| Aluno | Transmissão requisição (segundos) | Tempo no servidor (segundos) | Transmissão arquivo (segundos) | Tempo total (segundos) |
|---|---|---|---|---|
| Paulo | 1,48e-7 | 92,30 | 7 | 99,3 |
| Evandro | 7,278e-7 | 41,02 | 167,77 | 208,80 |
| Fábio | 8e-8 | 16,41 | 0,67 | 17,08 |
| Tânia | 1,11e-6 | 10,25 | 8,16 | 18,41 |

## 11.6 Exercícios

1) O aluno Evandro, dos exemplos do capítulo, está muito descontente com o desempenho da transmissão das aulas para seu computador. Ele quer uma sugestão de qual rede de

comunicação utilizar. Calcule o tempo de transmissão do arquivo de aula que ele deseja assistir para outros tipos de rede e sugira qual rede utilizar.

2) O aluno Paulo gostaria de saber quanto tempo ele levará para começar a assistir aos arquivos das aulas disponíveis no servidor. Calcule o tempo total que ele gastará para pedir e receber os arquivos em sua máquina, para cada tipo de aula.
3) Um website de uma agência de turismo foi monitorado por 30 minutos e 6000 requisições HTTP foram contadas. O servidor enviou três tipos de objetos: páginas estáticas HTML, imagens e vídeos. Os documentos HTML eram 50% das requisições com um tamanho médio de 15 Kbytes; imagens, 35% das requisições com tamanho médio de 30Kbytes; vídeos representaram 15% do número total de requisições, com tamanho médio de 1 Mbyte. Qual a vazão do servidor?
4) Um website de e-commerce recebe em média 20 visitantes por segundo. A maioria dos visitantes apenas conecta e lê as páginas HTML com ofertas, mas 10% dos visitantes fazem pedido de compra de livros. Cada transação de compra aciona um Servlet no servidor. Deseja-se saber qual é a utilização da CPU do servidor pelos servlets, sendo que o servidor possui a demanda por scripts de 300ms.
5) Uma empresa de treinamento está considerando o uso de vídeos para melhorar as aulas de treinamento. O tamanho médio de um vídeo de treinamento é 350 MB. As classes têm 100 estudantes, mas só 70% deles são ativos ao mesmo tempo. Durante a aula, cada aluno assiste, em média, a um vídeo por hora. Qual a largura de banda mínima necessária para esse tipo de aula?

# Avaliação de Desempenho com Virtualização 12

Para entender melhor o princípio da virtualização é preciso voltar à década de 1960, onde as camadas de hardware e do sistema operacional eram fortemente acopladas. Naquela época, as vendas eram sempre casadas entre o hardware e o software que iria rodar nele. Obviamente a comunidade de usuários não gostava disso e iniciou-se um movimento para desatrelar esses "gêmeos siameses". A separação teve início com um projeto da IBM que permitia rodar múltiplos ambientes de sistemas operacionais em um mesmo hardware, que era muito caro, diga-se de passagem. Surgiu, então, a ideia de VMM (*Virtual Machine Monitor*), que é a camada de software que provê a virtualização, mas que acabou perdendo fôlego na década de 1980, quando o hardware se tornou mais barato e os sistemas operacionais ganharam mais poder. A proposta foi retomada em um projeto de pesquisa de Stanford, em 1996, que viabilizou o uso de máquinas virtuais em hardware barato. Nesse momento, o desafio ainda era o de desatrelar o sistema operacional da arquitetura usada, o que foi feito com o uso de uma camada fina de software situada entre o hardware e o sistema operacional, virtualizando e gerenciando todos os recursos do hardware. Vale ressaltar que o princípio NÃO é o mesmo usado em ambientes Java.

Existem muitos conceitos para virtualização, alguns controversos. Todavia, neste livro, usaremos virtualização como um método de particionamento de um computador físico em múltiplos servidores virtuais, dando a cada um a capacidade de rodar em sua própria máquina dedicada. Cada servidor virtual funciona como um hardware com plenos direitos, podendo, inclusive, ser reiniciado de forma independente.

Baseados nesse princípio, pode-se supor que, havendo recursos suficientes na máquina nativa do usuário, vários sistemas podem conviver harmônica e simultaneamente em um mesmo ambiente (mesma máquina), delimitados por janelas conforme ilustrado na Figura 12.1.

Virtualizar um sistema operacional traz grandes vantagens e algumas desvantagens. Algumas das principais vantagens:

- Pode-se salvar o estado de um sistema virtualizado. Com isso, se você está configurando um servidor Linux, usando o editor *might command*, alterando a linha 10 do arquivo smb.conf e desejar sair rapidamente, é possível guardar esse estado e, quando o sistema for novamente ligado, pode-se retornar exatamente para esse ponto.
- Existe a possibilidade de recuperação rápida, uma vez que o sistema inteiro pode ser copiado (backupeado) e restaurado em caso de algum sinistro.

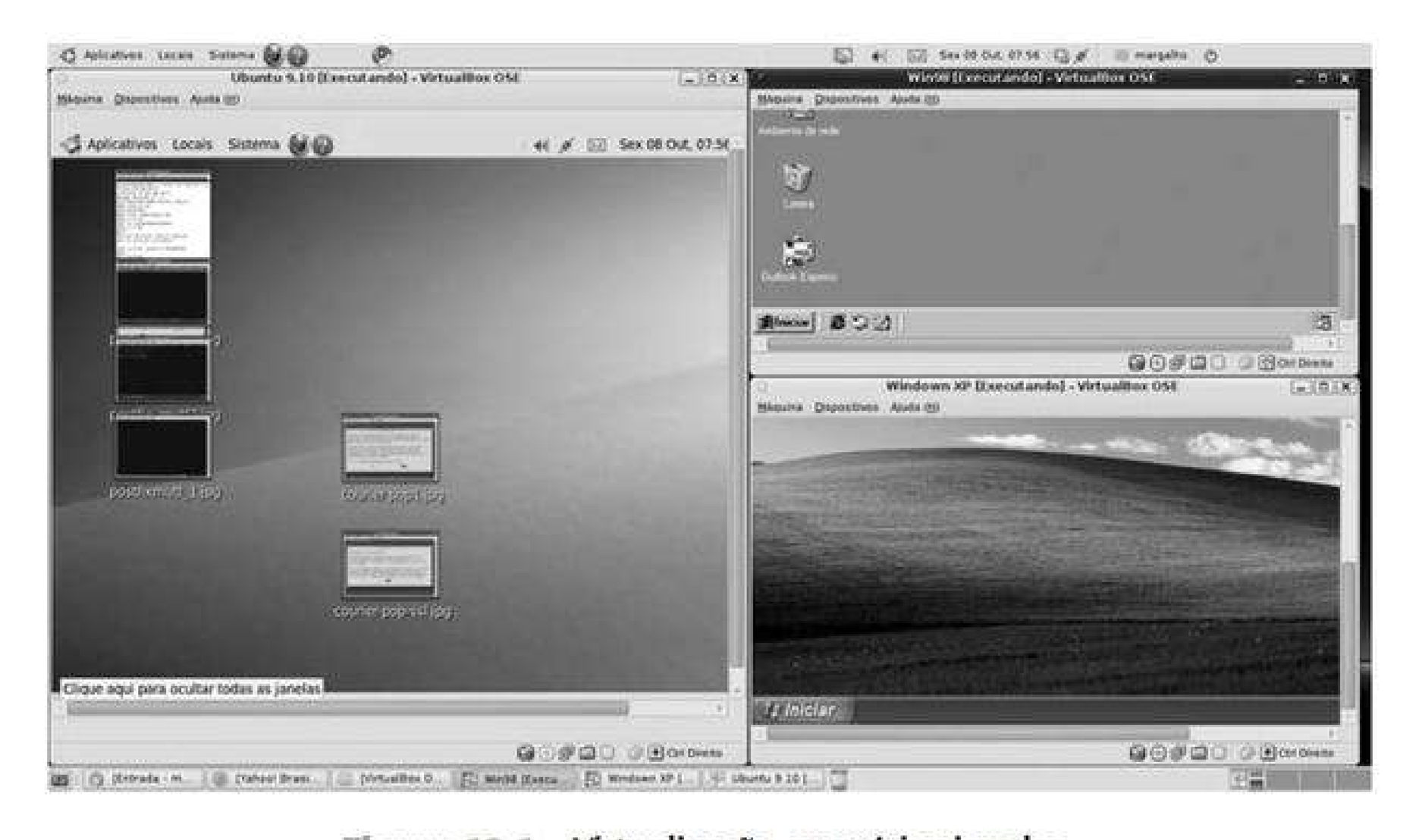


**Figura 12.1** Virtualização em várias janelas.

- O espaço físico requerido para acomodar um amontoado de servidores é minimizado consideravelmente, facilitando a própria gerência pelos administradores do sistema.
- Em ambientes educacionais, como, por exemplo, em um curso de redes de computadores, é possível permitir ao aluno a interação com diferentes Sistemas Operacionais de Rede ou NOS (*Netware Operating Systems*) a partir de um único hospedeiro (hardware físico). Alguns autores chamam essa convivência de "conformidade ideal".
- Existe uma racionalização na manutenção. Uma vez que haverá poucas máquinas físicas, a manutenção das mesmas também torna-se mais factível.
- Quebra-se a relação 1:1 entre aplicações e serviços e servidores.

Como desvantagens, pode-se elencar dois itens descritos:

- A segurança, pois se o mecanismo de virtualização ou VMM (*Virtual Machine Monitor*)[1] apresentar alguma vulnerabilidade, todas as máquinas virtuais que estão hospedadas nessa máquina física estarão vulneráveis.
- O desempenho, já que atualmente, não existem métodos consolidados para medir o desempenho de ambientes virtualizados. No entanto, a introdução de uma camada extra de software entre o sistema operacional e o hardware, o VMM ou hypervisor, gera um custo de processamento superior ao que se teria sem a virtualização. Outro ponto importante de ressaltar é que não se sabe exatamente quantas máquinas virtuais podem ser executadas por processador, sem que haja prejuízo da qualidade de serviço.

Alguns exemplos de softwares de virtualização gratuitos são o VMware Server da VMware (Figura 12.2), o VirtualBox da Sun (Figura 12.3) e o Virtual PC da Microsoft (Figura 12.4).

[1] VMM é uma camada de software e, por conseguinte, está sujeita a vulnerabilidades como qualquer software.

Neste capítulo será realizada uma avaliação bem simples para se medir o grau de perda de desempenho em uma máquina virtual. Para isso, serão utilizados dois computadores chamados

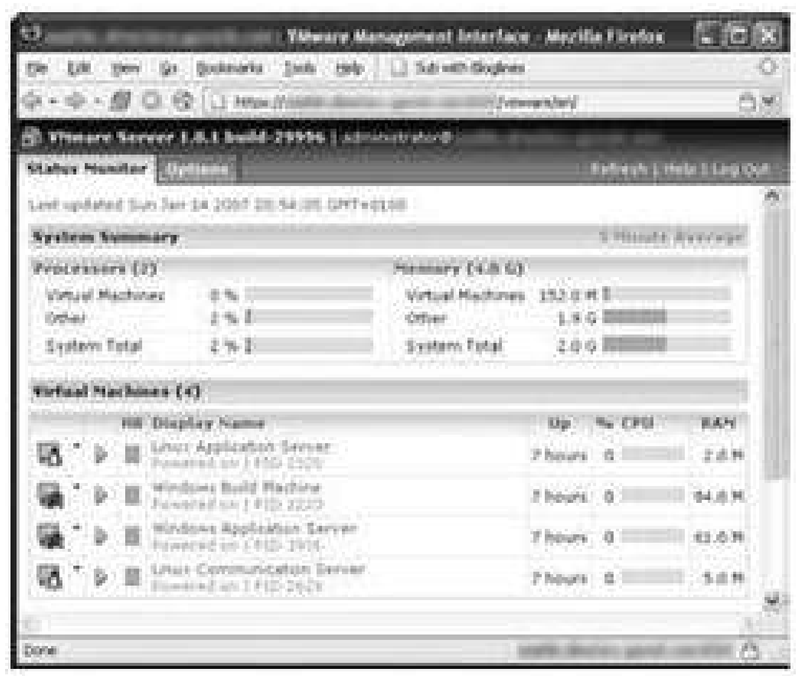


**Figura 12.2** VMware Server.

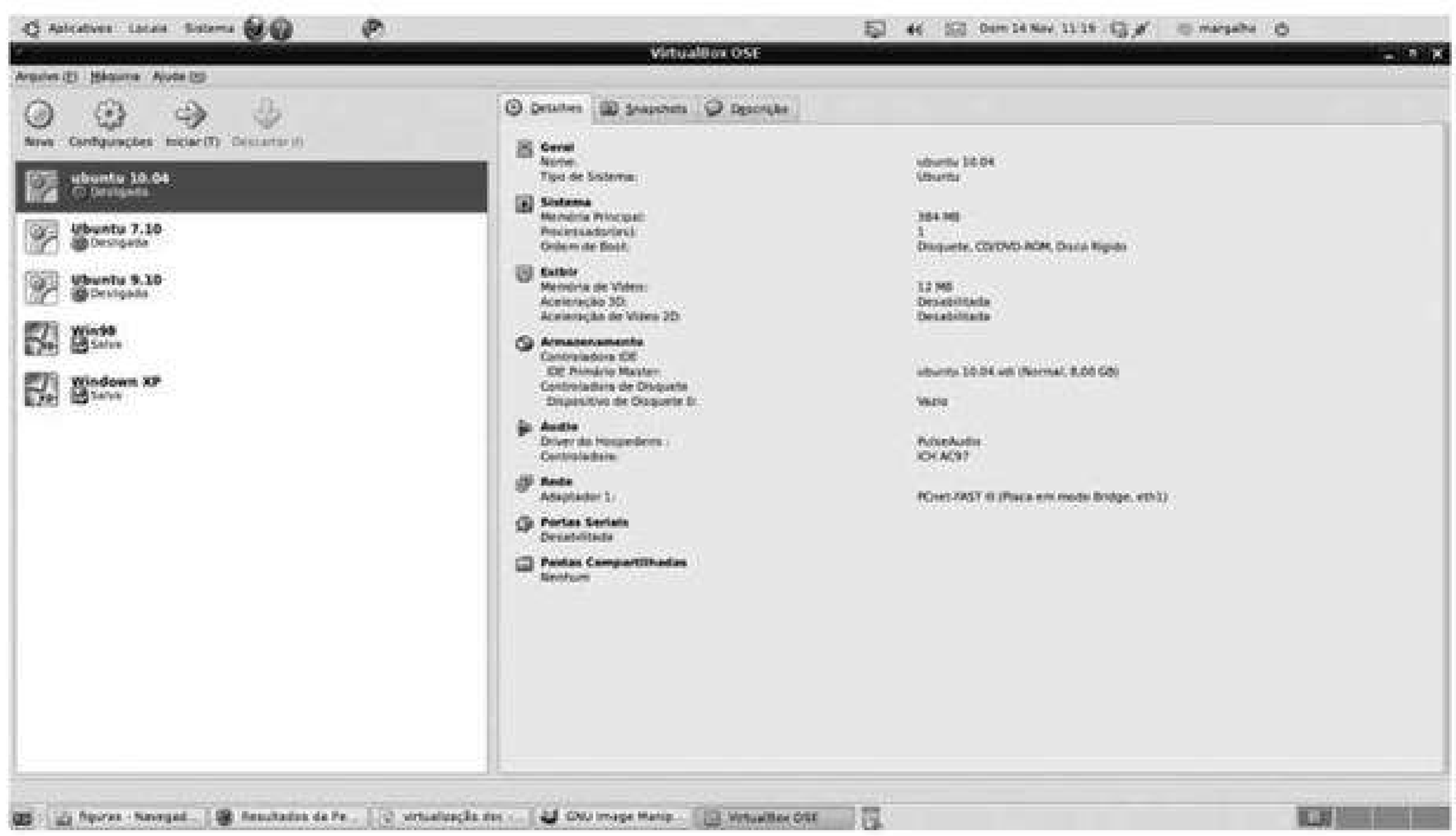


**Figura 12.3** VirtualBox.

**Figura 12.4** Virtual PC.

de Host1 e Host2 com o mesmo sistema operacional instalado (neste exemplo, utilizou-se o Linux Ubuntu Desktop LTS versão 10.04). Dentro de cada um dos Hosts foi instalado o software de virtualização VirtualBox da Sun e novamente o mesmo sistema operacional dos Hosts. Essa disposição pode ser visualizada na Figura 12.5.

Há de se considerar que as máquinas foram integralmente dedicadas ao processo de avaliação sem que nenhum software adicional estivesse sendo executado durante o experimento.

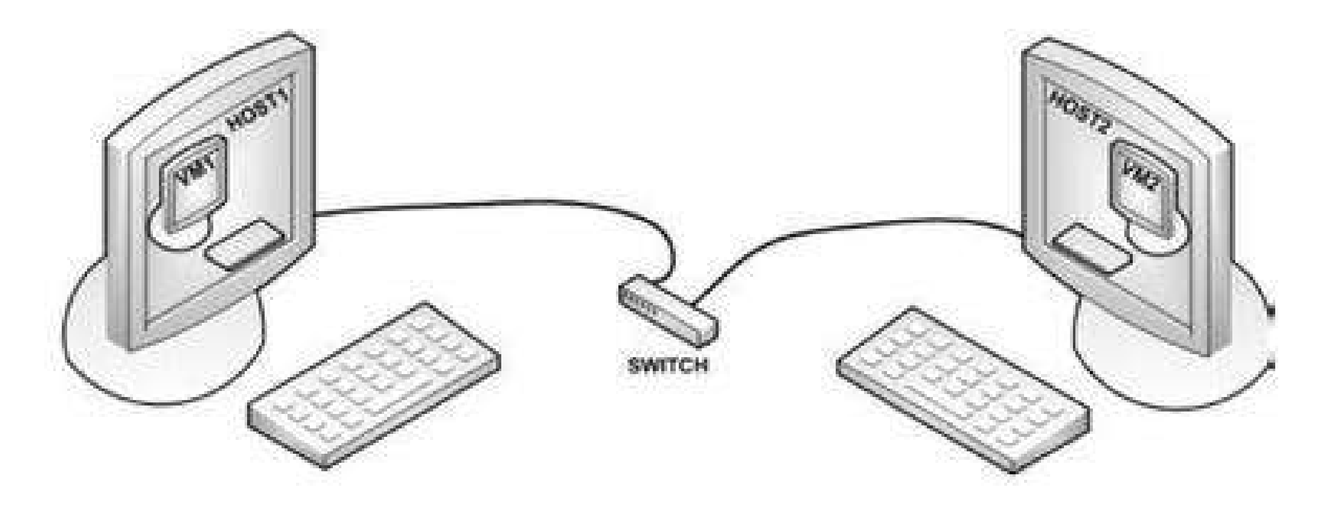


**Figura 12.5** Cenário da virtualização.

Após as quatro máquinas estarem interconectadas por uma estrutura de rede local, utilizou-se o software IPERF com as mesmas configurações apresentadas no Capítulo 9 (Quadro 12.1), à exceção da taxa a ser injetada no enlace, que foi definida em 100 Mb. Foram medidas a taxa de transferência e o *jitter* entre os Hosts externos (Host1 e Host2) e entre as máquinas virtuais (VM1 e VM2).

O resultado observado no gráfico da Figura 12.6 mostra a diferença na taxa efetiva de transferência entre as máquinas externas (Host1 e Host2) e as máquinas virtuais internas (VM1 e VM2).

**QUADRO 12.1** Configuração dos hosts

| Hosts | |
|---|---|
| **Servidor IP 10.0.0.69/8** | **Cliente IP 10.0.0.42/8** |
| **iperf –s –u –i (intervalo de apresentação) > arquivo.txt** | **iperf –c (IP do servidor) –u –b 100 m –t (tempo para medição)** |
| –s = servidor | –c = cliente |
| –u = udp | –u = udp |
| –i = intervalo no qual os dados lidos são exibidos em segundos (10) | –b = velocidade (carga) a ser injetada no link |
| | –m = Formato de megabits |
| | –t = tempo de medição em segundos (900) |

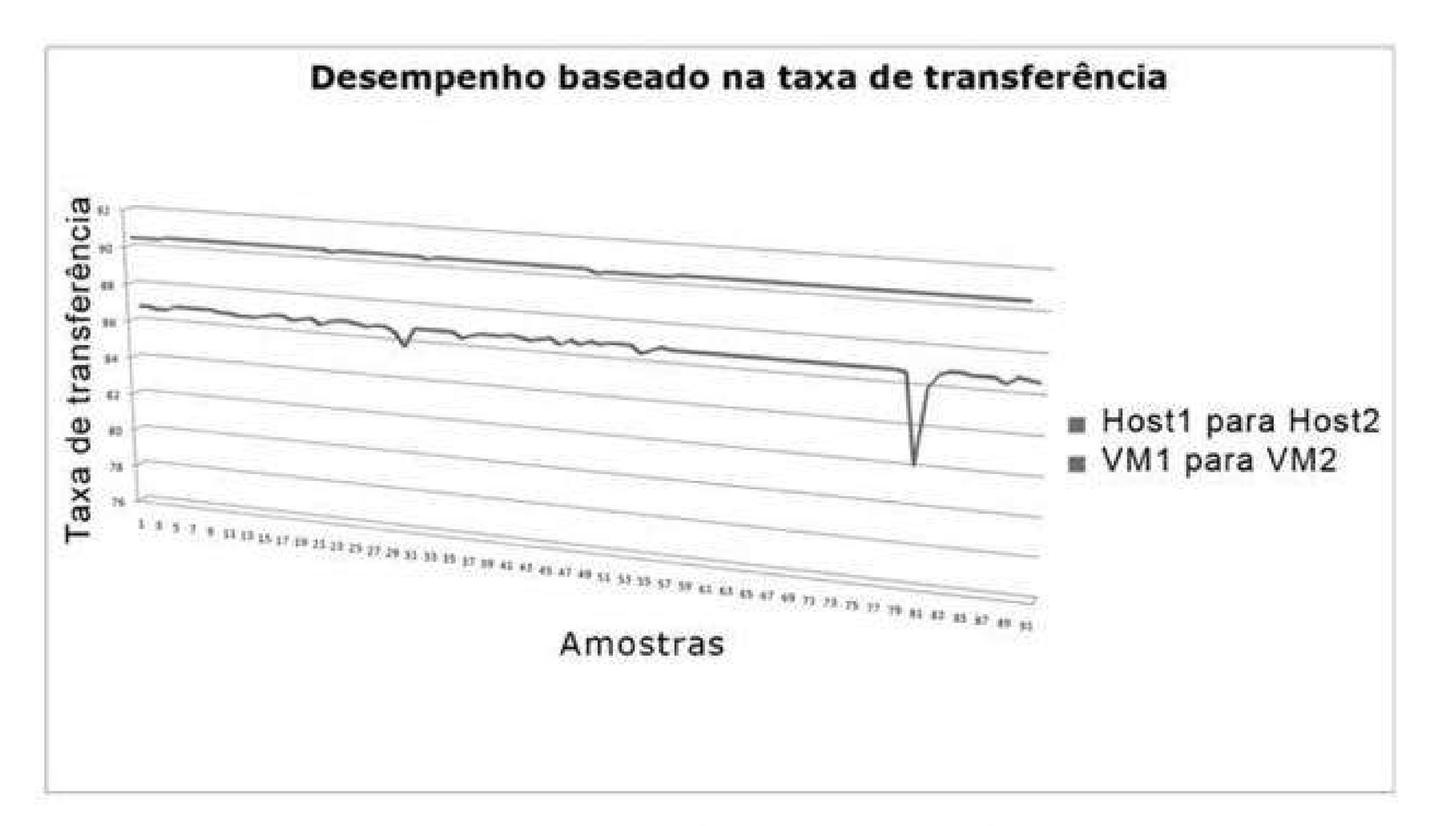


**Figura 12.6** Desempenho baseado na taxa de transferência.

No gráfico das médias, apresentado na Figura 12.7, pode-se observar mais claramente essa diferença, onde a transferência de dados entre sistemas operacionais das máquinas nativas (Host1 e Host2) possui um desempenho melhor do que nas máquinas virtualizadas.

Já no gráfico da Figura 12.8, que representa o *jitter*, a situação se inverte. Isso ocorre porque quanto menor o *jitter*, melhor o desempenho das aplicações. Variações grandes dessa métrica podem, inclusive, inviabilizar aplicações multimídia.

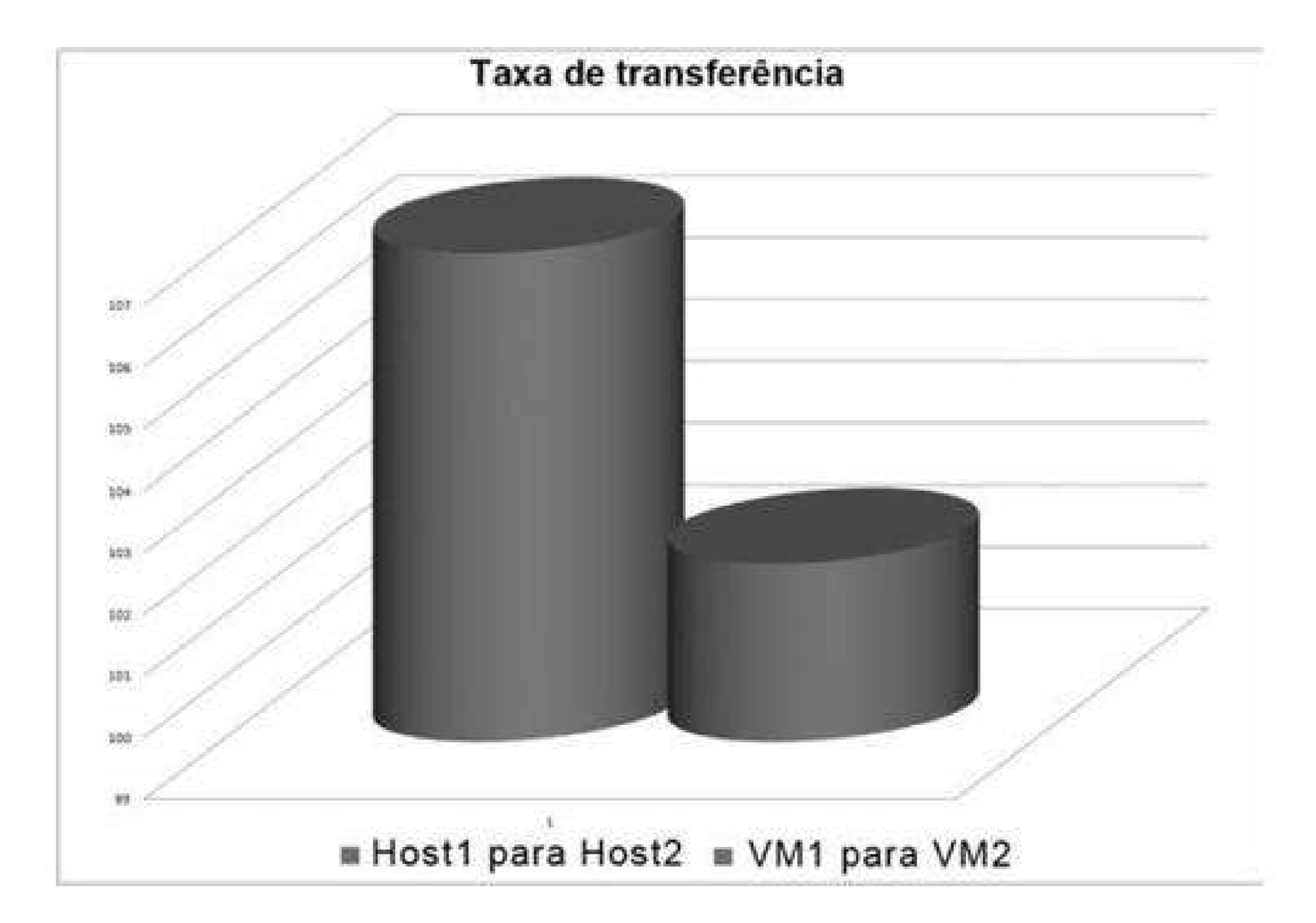


**Figura 12.7** Média das taxas de transferência.

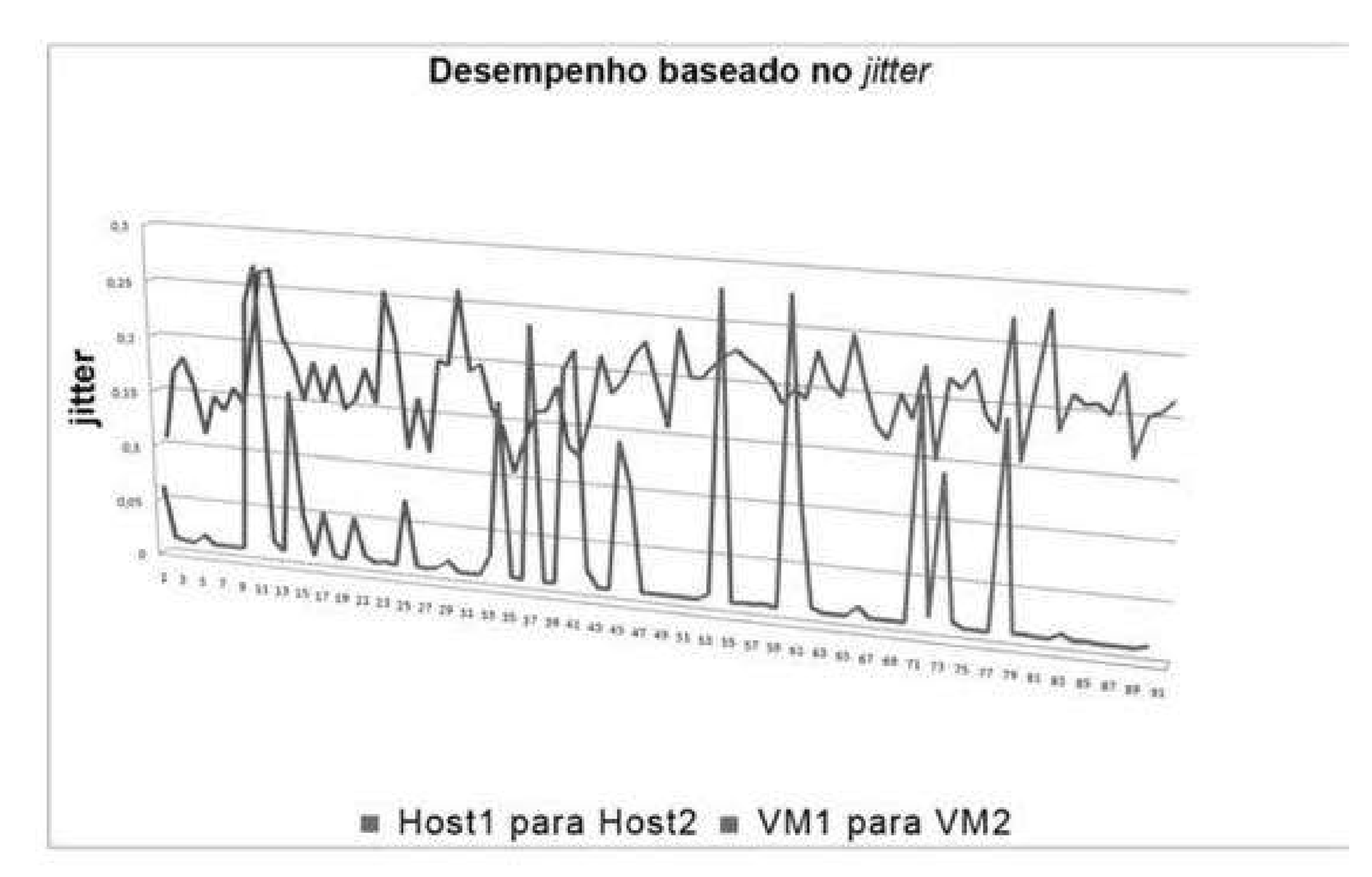


Figura 12.8 Desempenho baseado no *jitter*.

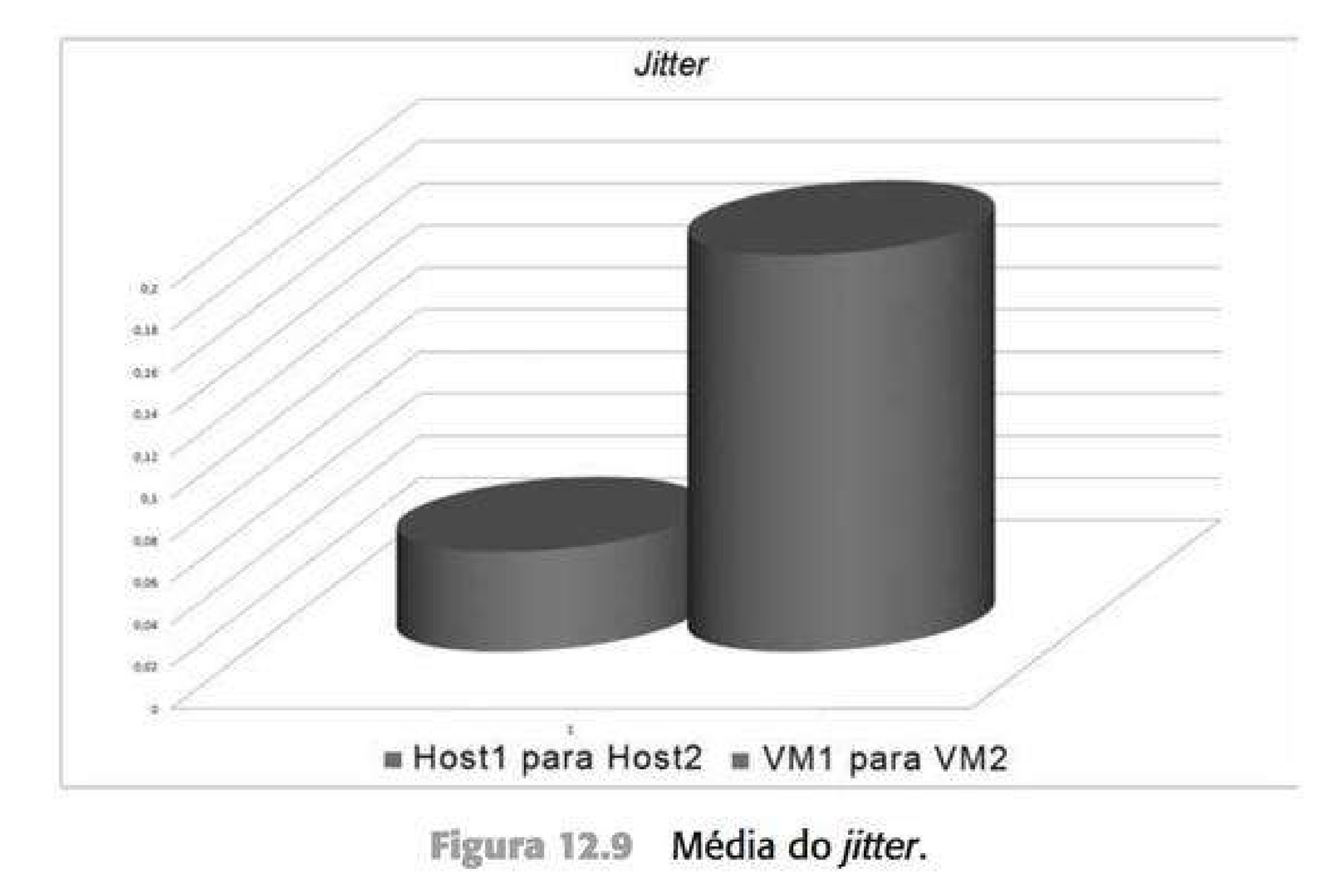


Figura 12.9 Média do *jitter*.

## 12.1 Exercícios

Como atividade complementar, sugere-se que seja reproduzido todo o experimento aqui descrito. Todavia, deve-se acrescentar a medição entre o Host1 e a máquina virtual 2 (VM2). Depois de concluída a medição, crie os gráficos e faça o comparativo.

# Simulação com Redes de Sensores 13

Sensores são dispositivos que coletam continuamente dados a partir de uma reação a estímulos físicos como, por exemplo, movimento ou luz. Em uma rede de sensores típica, que pode ser definida como uma distribuição de sensores autônomos que trabalham para monitorar condições físicas e ambientais, há um trabalho colaborativo com o objetivo de fazer com que os dados coletados cheguem a um nó específico chamado de sorvedouro. Para melhor compreensão do funcionamento de uma rede de sensores, deve-se iniciar pelas topologias usadas nesse ambiente.

## 13.1 Topologias de Redes sem Fio

As duas principais topologias de redes sem fio são a infraestruturada e a ad-hoc. Na topologia infraestruturada, ilustrada na Figura 13.1, toda e qualquer comunicação entre os nós (clientes) da rede passa sempre por um ponto centralizado. A grande vantagem desse modelo é a facilidade de gerenciamento. Uma vez que toda a comunicação passa sempre por um ponto que pode ser monitorado, é possível a realização de procedimentos de bilhetagem e contabilização dos recursos, facilitando a administração. Um exemplo de rede que usa esse modelo de topologia é a rede da telefonia celular, que tem seus pontos de centralização baseados nas ERBs (Estações Rádio Base). Devido às características da topologia, mesmo que uma pessoa esteja a poucos metros da outra, não haverá comunicação caso não seja estabelecida uma conexão com uma ERB para intermediar esse processo.

Já em uma topologia ad-hoc, ilustrada na Figura 13.2, não há essa restrição para a comunicação, ou seja, os nós podem assumir uma de três funções: transmissor, roteador ou receptor. A grande vantagem desse tipo de rede é justamente o trabalho colaborativo que existe para entregar os dados no destino. Se o nó destino não estiver na área de cobertura do nó que está transmitindo, outros nós intermediários podem pegar esses pacotes, injetar potência[1] e retransmiti-los em um procedimento cíclico, até que o destino seja alcançado. Obviamente essa vantagem tem consequências que se refletem na maior dificuldade de gerenciamento da rede.

A sequência de imagens apresentada nas Figuras 13.3 a 13.11 — sequência de colaboração ad-hoc — mostra claramente a colaboração entre os nós com o objetivo de atingir o sorvedouro. Essas imagens refletem o cenário que será considerado no estudo de caso deste capítulo e podem ser consideradas

[1] O termo injetar potência serve apenas para ilustrar a ação de um nó receber o sinal fraco e retransmiti-lo, permitindo assim um alcance maior, se for tomado como referência a origem da transmissão.

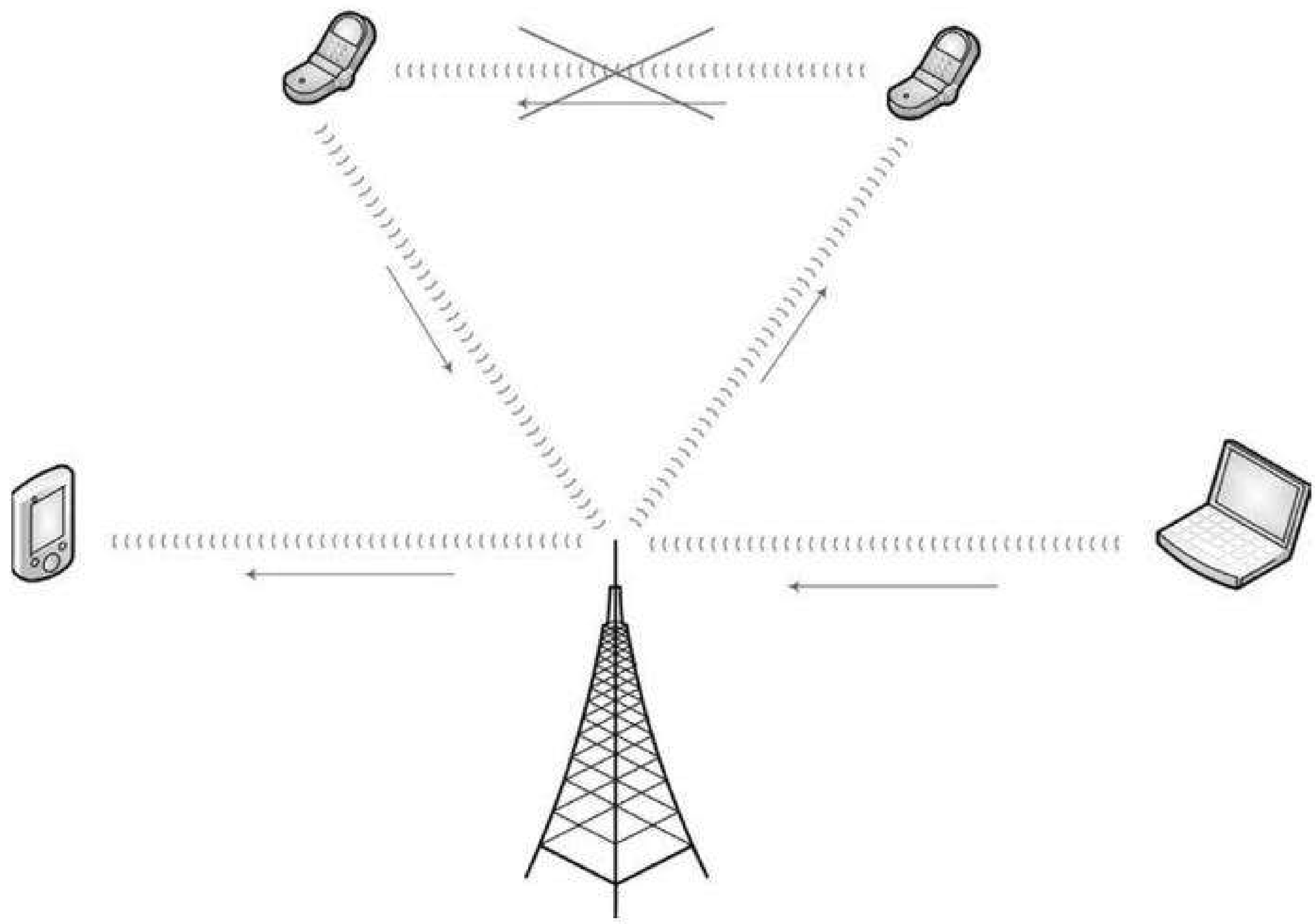

Figura 13.1 Topologia infraestruturada.

fotos instantâneas ou *snapshots* tiradas em alguns instantes de simulação. Inicialmente, os sensores trocam entre si informações de sinalização que serão usadas no estabelecimento das rotas (Figura 13.3). Em seguida, o sensor 1, representado pelo nome de câmera 1, inicia a transmissão diretamente ao sorvedouro (Figura 13.4). Com a movimentação do sensor 1 as dificuldades de comunicação começam a surgir e faz-se necessário o apoio do sensor 2 para que os dados cheguem ao sorvedouro (Figura 13.6). Conforme o sensor 1 se desloca, mais mudanças de rota ocorrem. Na Figura 13.8 houve a intermediação do sensor 4 para viabilizar a comunicação. Na Figura 13.9 essa intermediação foi do sensor 3, e na Figura 13.11, dos sensores 3 e 5. Os pontos mais escuros, nas figuras intermediárias, indicam a ocorrência de descartes de pacotes, o que requer, normalmente, um ajuste de rota.

Neste capítulo será explorado, portanto, o uso de redes de sensores em simulação no NS-2. Para facilitar o entendimento do leitor, todo procedimento de modelagem via simulação será explicado a partir de um cenário contextualizado.

## 13.2 Contextualização do Cenário

Diante de tantas notícias de assaltos onde os computadores que mantêm as imagens das câmeras de segurança são levados ou destruídos pelos meliantes, a empresa "Fictícia S.A." solicitou a

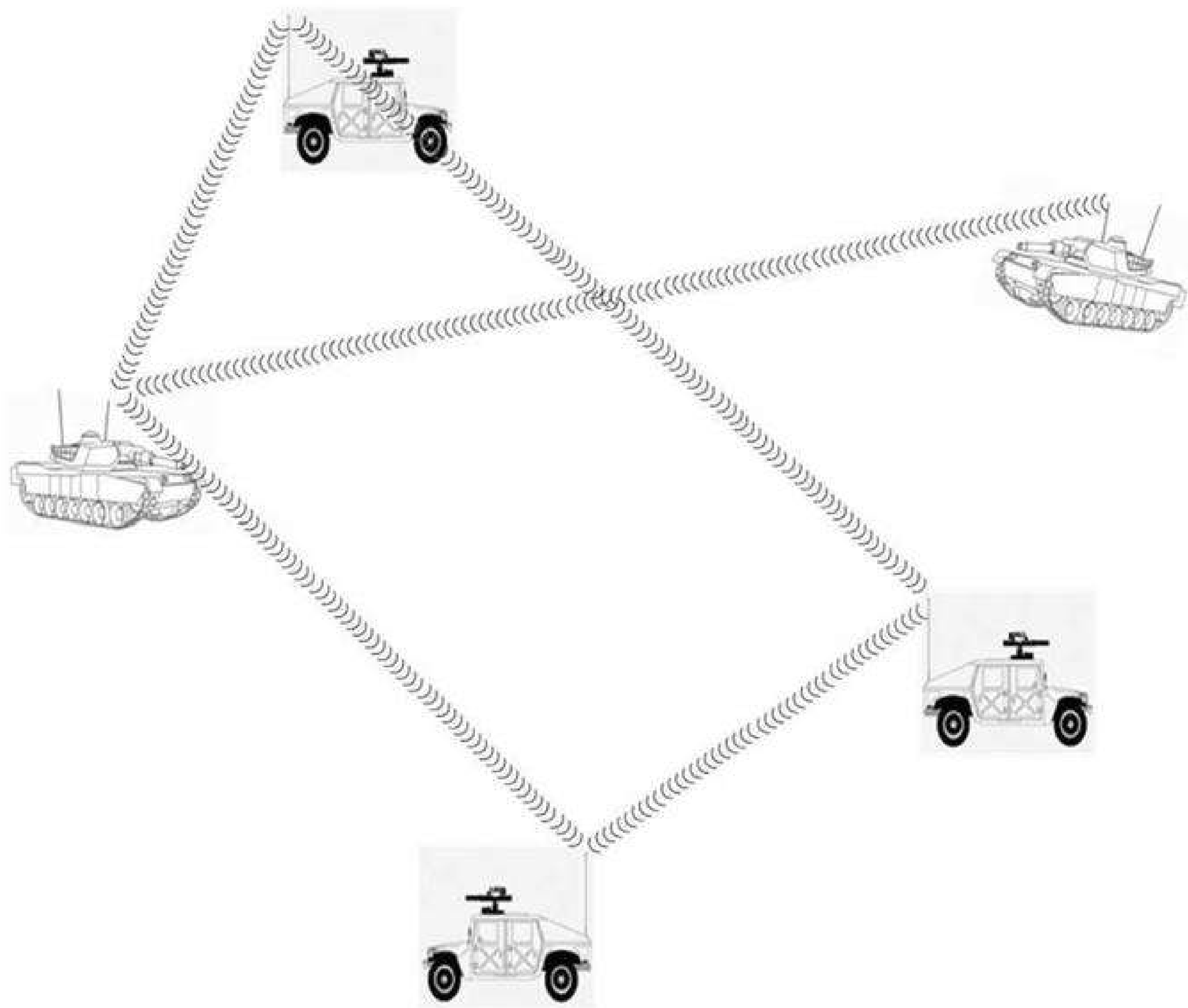

**Figura 13.2** Topologia ad-hoc.

instalação de um sistema mais seguro, que transmita e armazene as imagens em um ponto remoto. Para isso, o CTIC (Centro de Tecnologia da Informação e Comunicação) da empresa apresentou

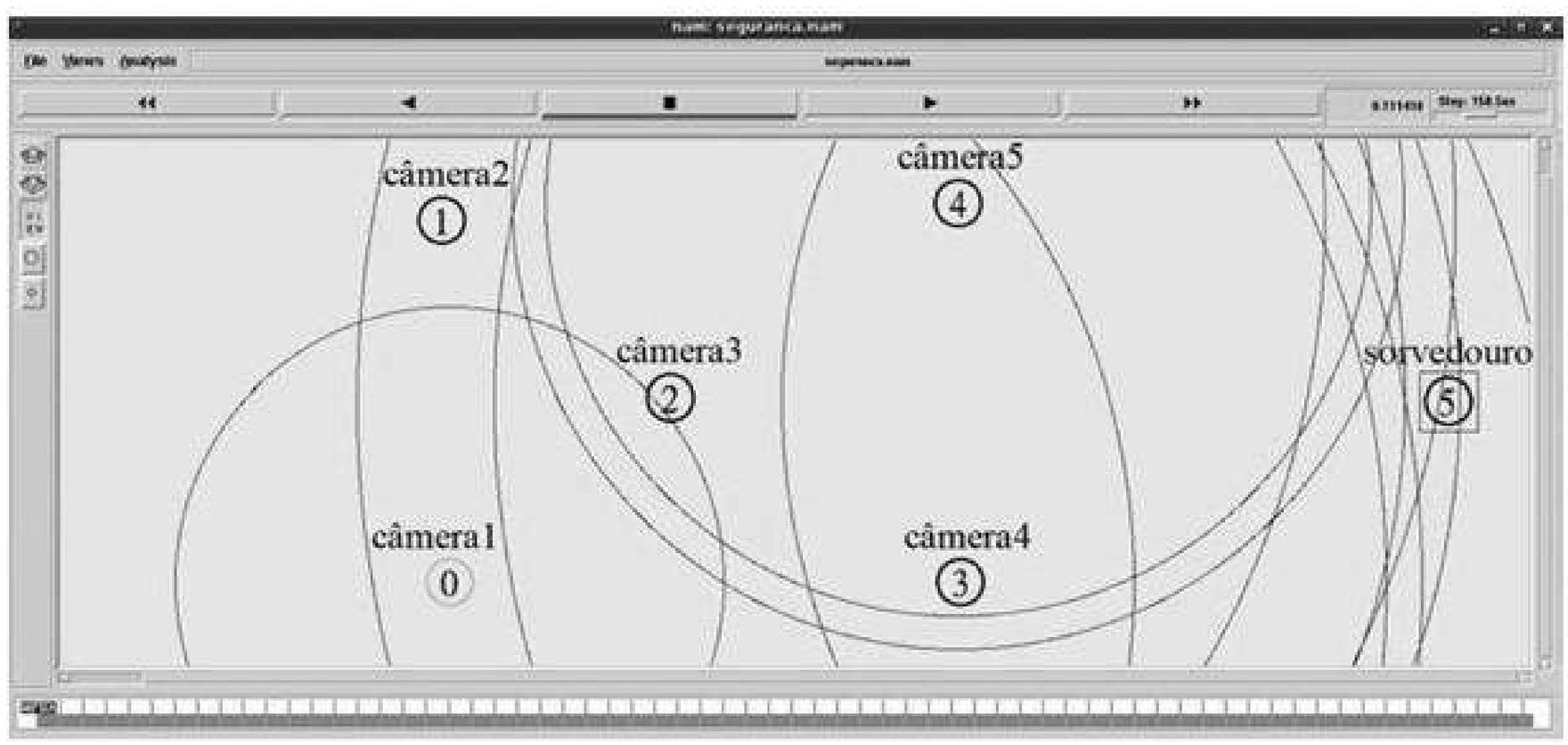


**Figura 13.3**

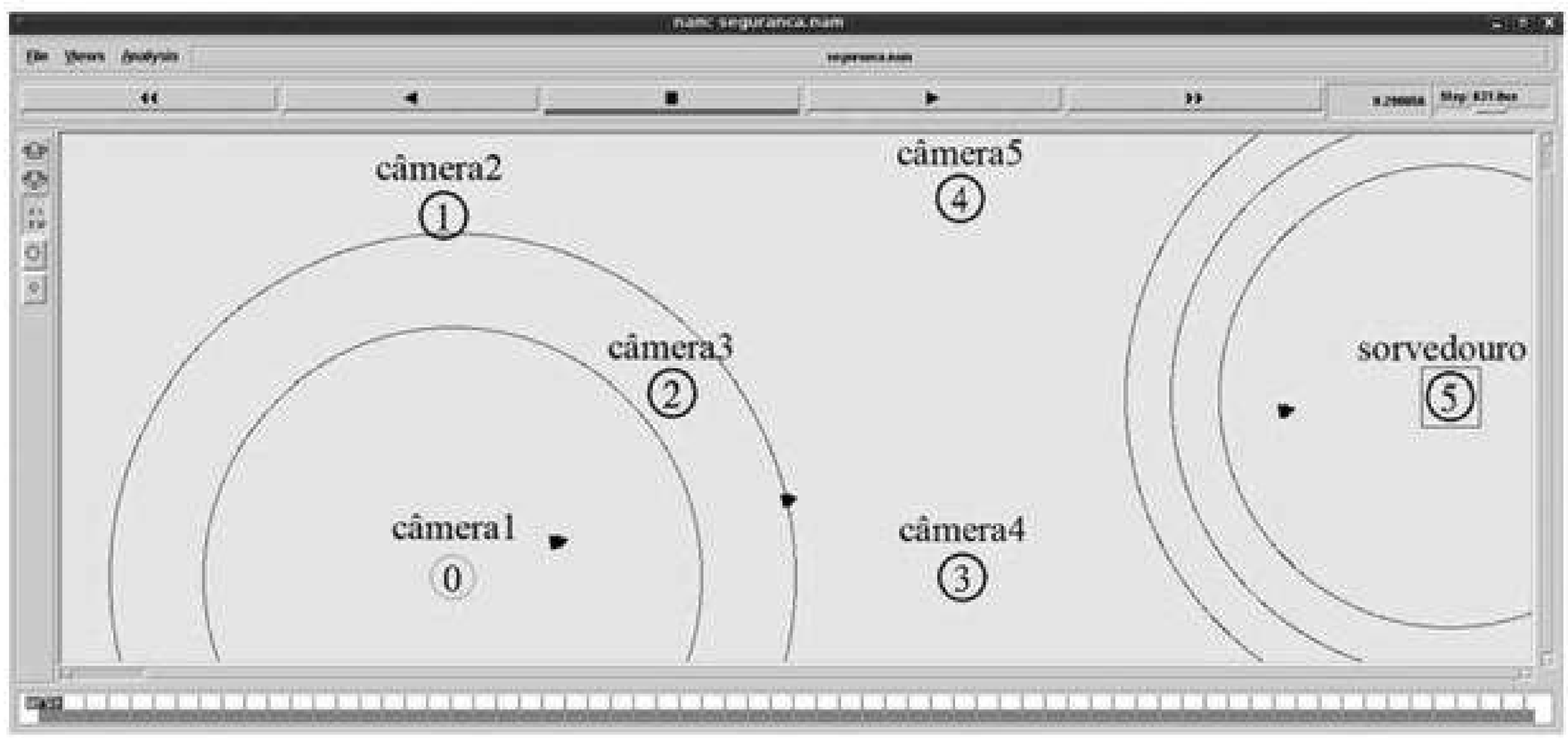


Figura 13.4

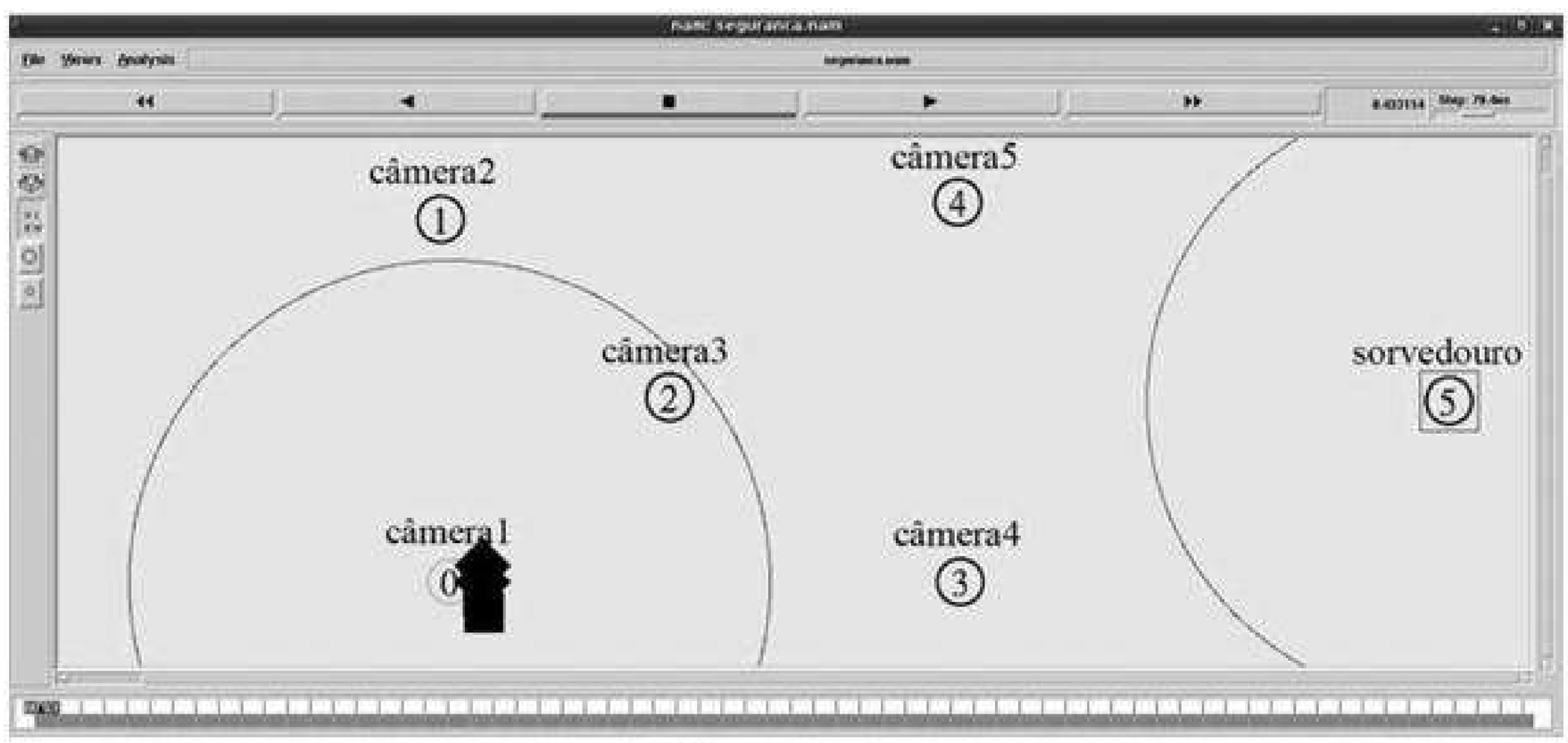


Figura 13.5

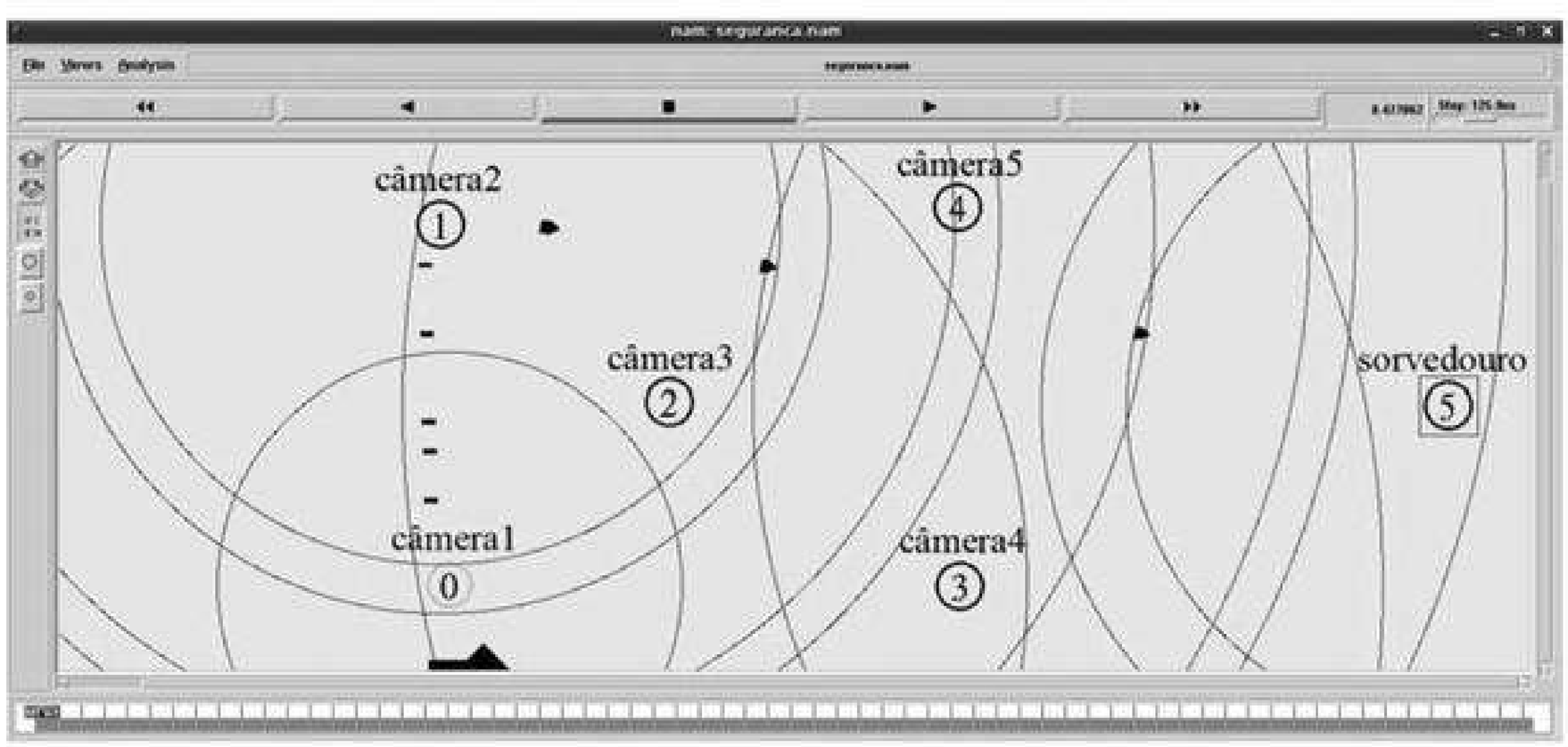


Figura 13.6

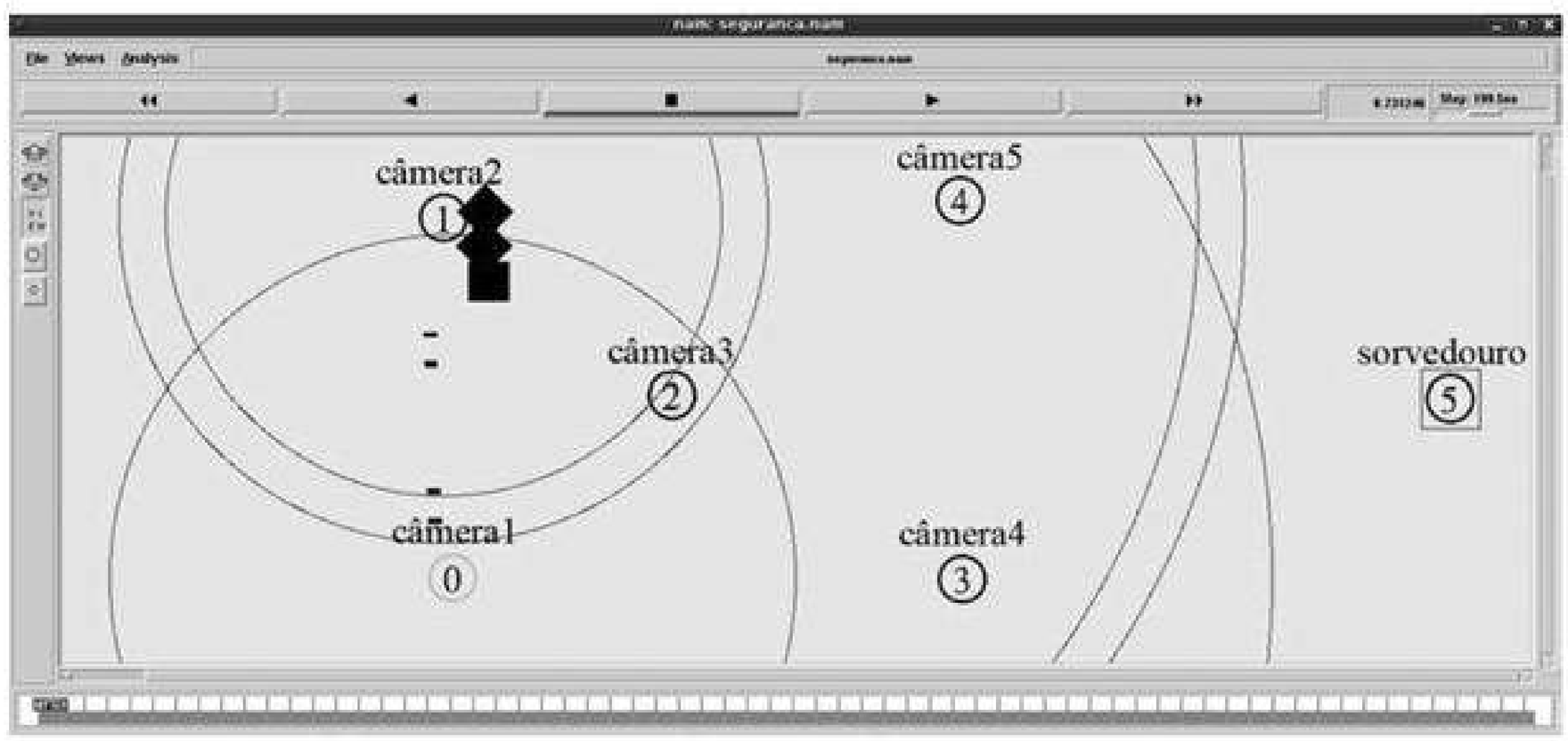


Figura 13.7

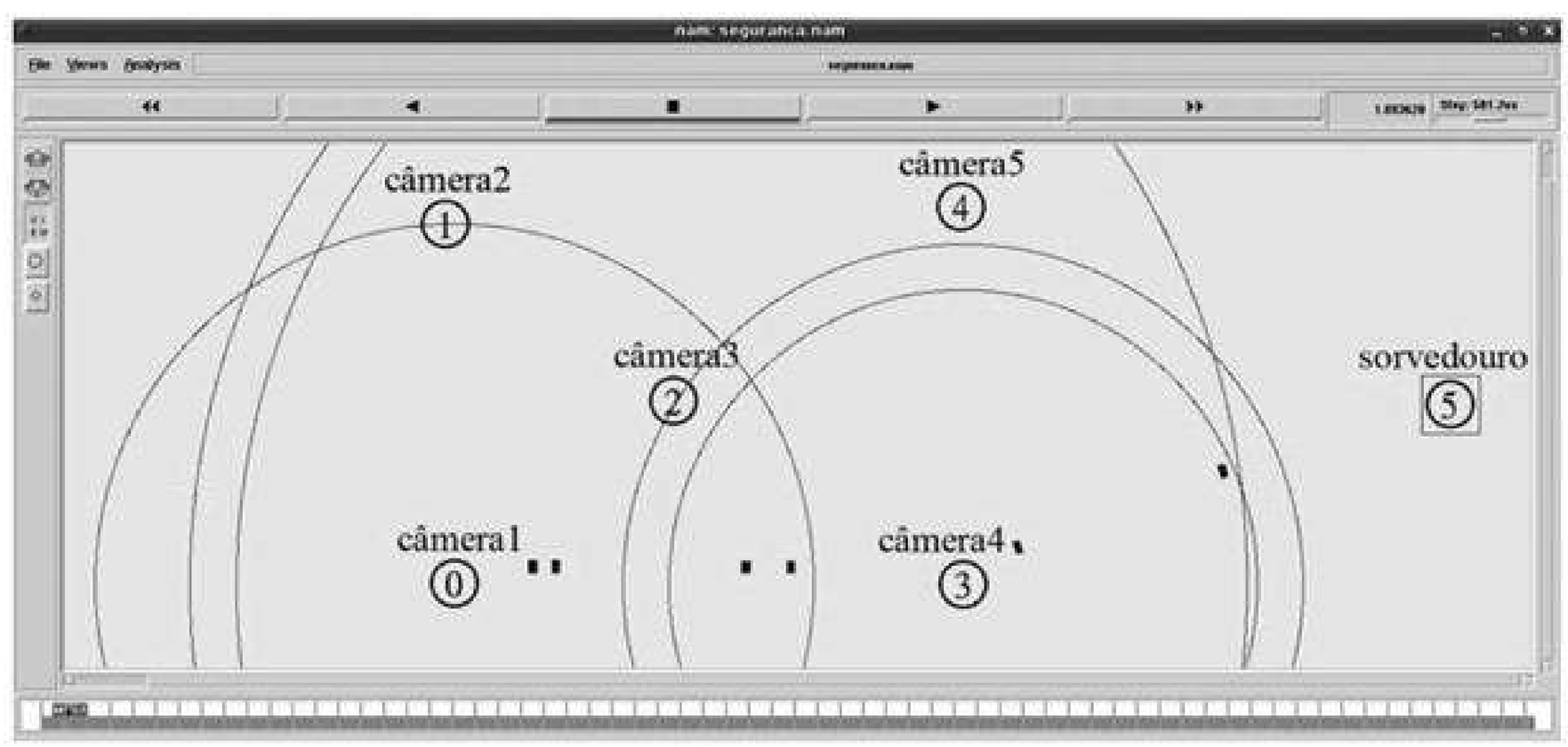


Figura 13.8

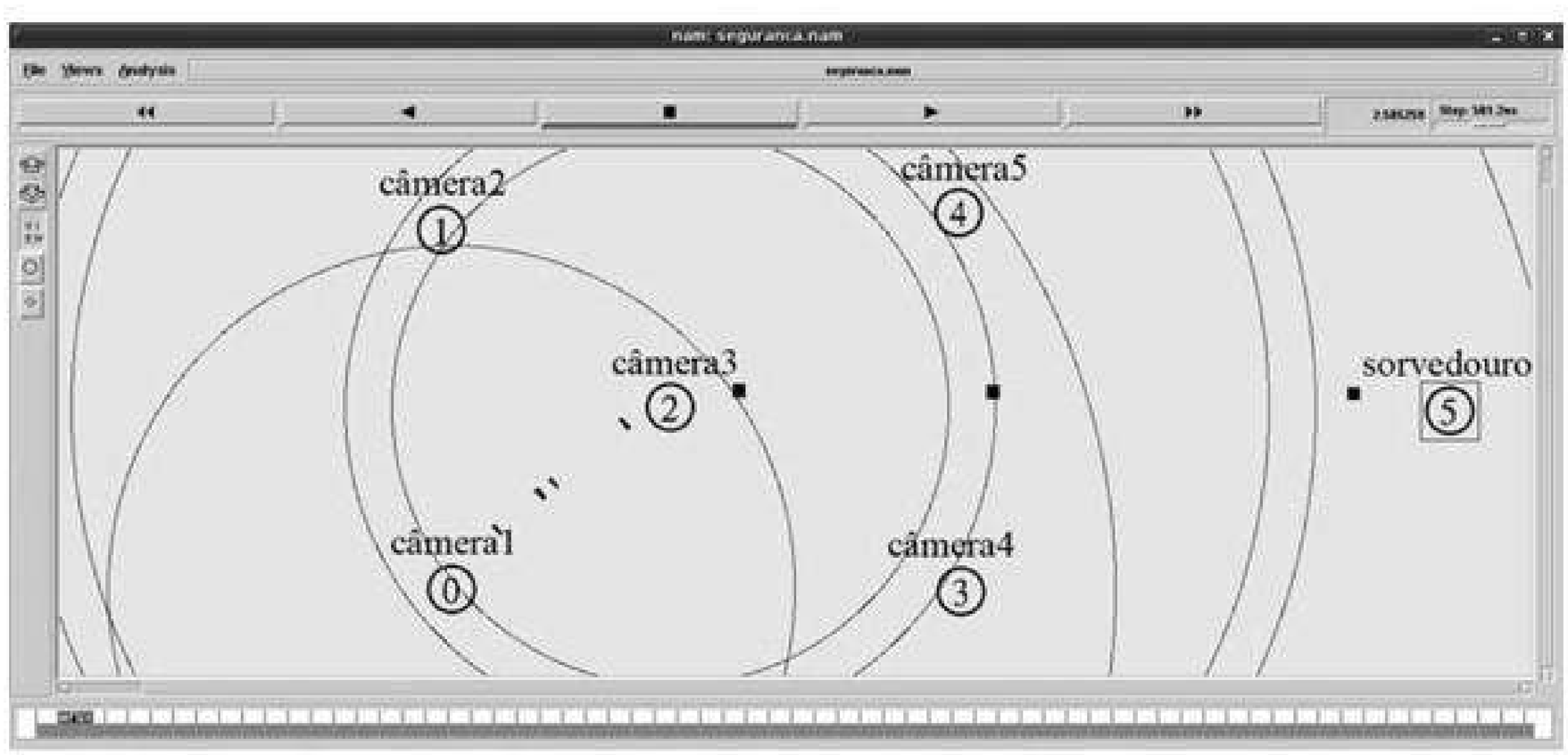


Figura 13.9

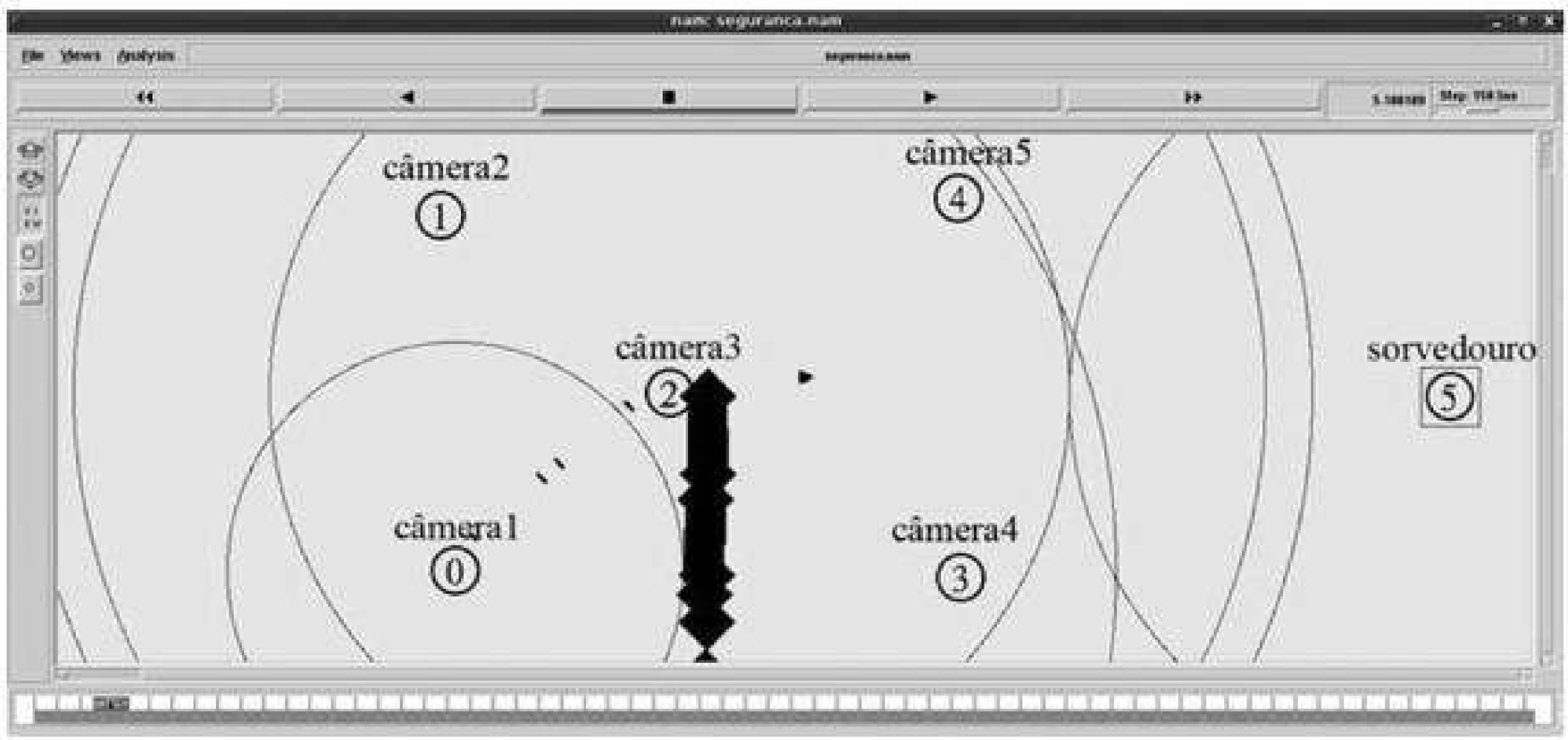


Figura 13.10

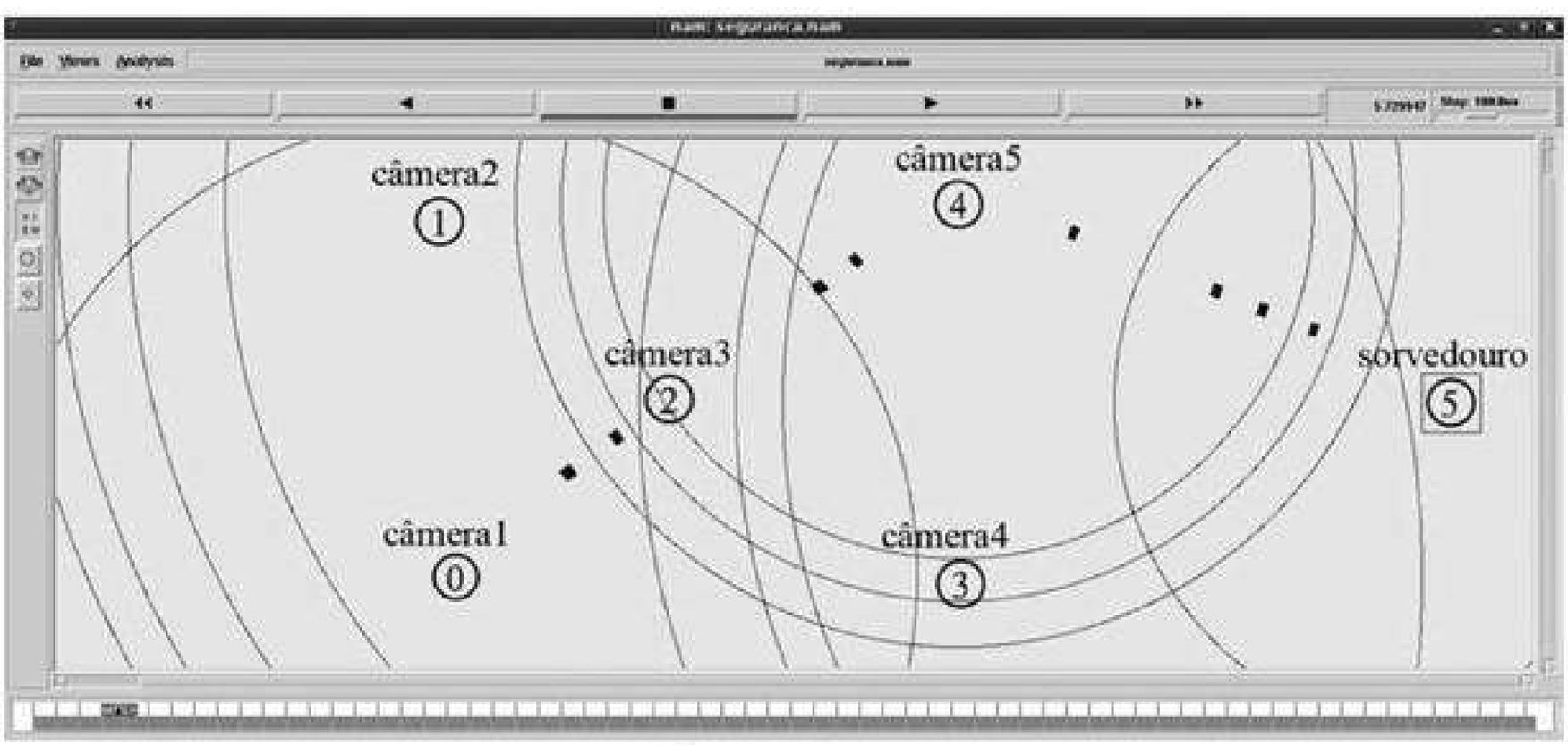


Figura 13.11

uma proposta onde seriam instalados sensores de movimento acoplados a câmeras e posicionados em pontos estratégicos de forma que, ao detectarem qualquer movimento, os sensores ativam as câmeras que iniciam imediatamente a transmissão das imagens para o ponto remoto (sorvedouro). Além de transmitir as imagens, as câmeras acopladas aos sensores deslocam-se em um trilho suspenso procurando a melhor posição e ângulo para a captação das imagens.

## 13.3 Definição da Área de Localização dos Sensores

Para entender como o NS-2 trabalha com simulação de redes sem fio, o passo inicial é absorver o conceito de FlatGrid. Trata-se de uma área onde eventualmente pode ser feita alguma referência a um nó da rede. Pode-se visualizar o FlatGrid como um eixo cartesiano em que "X" representa a distância do nó no eixo das abscissas, "Y" representa a distância do nó no eixo das ordenadas e "Z" representa a altura. Todo e qualquer nó localizado nessa área precisa ser referenciado geograficamente. Para efeitos didáticos, pode-se utilizar uma unidade padrão, como o metro, por

exemplo. Neste cenário, considera-se que a área do FlatGrid é de 100 × 100 m. Isso significa que durante a simulação nenhum nó poderá ser referenciado na posição ($x = 101$; $y = 10$; $z = 10$). Se isso ocorrer haverá um erro no simulador uma vez que o FlatGrid somente prevê que os nós se desloquem até o ponto 100 no eixo das abscissas. A Figura 13.12 mostra os nós distribuídos no FlatGrid que caracteriza esta simulação.

À exceção do posicionamento e deslocamento dos nós, todos os conceitos apresentados nos Capítulos 6, 7 e 8 deste livro continuam válidos.

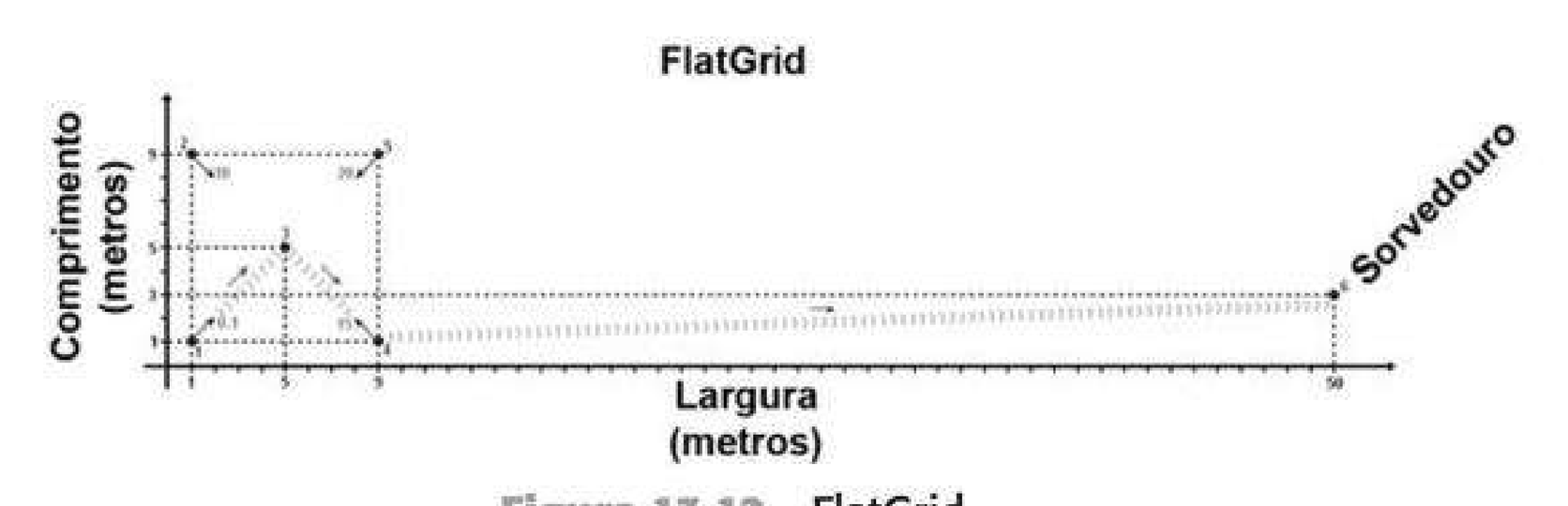


**Figura 13.12** FlatGrid.

## 13.4 Codificando a Simulação

Uma vez que os principais elementos de uma simulação com redes guiadas já foram abordados anteriormente neste livro, serão enfatizados aqui apenas os itens que caracterizam o ambiente sem fio. Para tanto, será usada a metodologia de agrupar e explicar os trechos do código.

**QUADRO 13.1** Configurações iniciais

| | | |
|---|---|---|
| 1 | # ========== | |
| 2 | # Simulacao de Redes de Sensores | Comentários |
| 3 | # ========== | |
| 4 | set wnossemfio 6 | Número de nós sem fio |
| 5 | set distx 100 | Comprimento da área em metros |
| 6 | set disty 100 | Largura da área em metros |
| 7 | set tempo_simulacao 100 | Tempo de simulação |
| 8 | set num_wired_nodes 0 | Número de nós cabeados |
| 9 | set num_access_points 0 | Número de pontos de acesso |

O Quadro 13.1 apresenta as configurações iniciais da simulação. Cabe lembrar que o caractere "#", disposto no início de uma linha, indica comentário. A linha 4 é usada para se criar uma variável chamada wnossemfio, que armazenará o número de nós sem fio. Neste exemplo serão utilizados cinco nós sensores e um sorvedouro, em um total de seis nós sem fio. A vantagem em se usar essa metodologia de programação é que, havendo necessidade de mudança no número de sensores, o ajuste será feito em um único lugar.

Para se fazer referência ao conteúdo de uma variável basta colocar "$" antes do nome da variável. Exemplo disso são as variáveis criadas nas linhas 5 e 6 do Quadro 13.1, chamadas distx e disty. Elas armazenam, respectivamente, o comprimento e a largura da área onde ocorrerá a simulação ou FlatGrig. No cenário adotado como exemplo, essa área é de 100 por 100 metros. Também foram definidas variáveis para o tempo total de simulação (linha 7), o número de nós cabeados (linha 8) e o número de pontos de acesso (linha 9).

**QUADRO 13.2** Parâmetros

| | | |
|---|---|---|
| 1 | # ========== | Comentários |
| 10 | # ========== | |
| 11 | #Parametros | |
| 12 | # ========== | |
| 13 | Antenna/OmniAntenna set Z_ 0.1 | Altura da antena |
| 14 | Antenna/OmniAntenna set Gt_ 12.0; | Ganho de transmissão |
| 15 | Antenna/OmniAntenna set Gr_ 12.0; | Ganho de recepção |
| 16 | Phy/WirelessPhy set bandwidth_ 2Mb | Largura de banda |
| 17 | Phy/WirelessPhy set Pt_ 0.031622777; | Potência de transmissão |
| 18 | Phy/WirelessPhy set Pr_ 0.2818; | Potência de recepção |
| 19 | Phy/WirelessPhy set freq _ 914e+6 | Frequência |
| 20 | Mac/802_11 set dataRate_ 2Mb | Taxa de transmissão |
| 21 | Mac/802_11 set basicRate_ 1Mb | Taxa básica |
| 22 | Mac/802_11 set PreambleLength_ 72 | Tamanho do preâmbulo |
| 23 | Propagation/Shadowing set pathlossExp_ 4 | Expoente de perdas |
| 24 | Propagation/Shadowing set std_db_ 5.0 | Desvio padrão |

O Quadro 13.2 apresenta os principais parâmetros da simulação. Uma simulação bem elaborada requer um estudo minucioso do valor a ser adotado em cada parâmetro. Um único equívoco nessa etapa pode comprometer totalmente os resultados. As linhas 13, 14 e 15 do Quadro 13.2 mostram os parâmetros a serem usados na antena *omni* direcional, instalada em cada um

dos sensores. Como em um sensor as antenas podem estar embutidas, considerou-se a altura como praticamente zero. As linhas 14 e 15 indicam, respectivamente, os ganhos de transmissão e recepção que devem ser pesquisados junto à especificação do fabricante. Se você não tiver essa informação pode considerar o ganho de uma antena isotrópica, que é 1. Na linha 16, deve-se informar qual a largura de banda associada ao padrão usado. Se, por exemplo, o padrão for de uma rede sem fio IEEE 802.11b, então esse valor seria de 11 Mbps. Caso fosse usado o padrão IEEE 802.11g, 54 Mbps e assim por diante. Neste exemplo, o valor usado foi de 2 Mbps.

Nas linhas 17 e 18 devem ser informadas, respectivamente, as potências de transmissão e recepção dos nós sensores, em watt. A linha 19 indica a frequência a ser usada. Nesse exemplo, usou-se uma notação científica, aceita pelo NS-2, para informar a frequência de 914 MHz. As linhas 20 e 21 mostram os limiares aceitáveis para que a conexão possa ser estabelecida. Isso ocorre porque, em situações onde há muita interferência, há uma negociação entre o transmissor e o receptor permitindo que seja estabelecida a conexão, mesmo com taxas de transmissão abaixo da capacidade do padrão adotado. Todavia, essa margem tem um limite. A linha 22 mostra o valor do preâmbulo que pode ser obtido junto à documentação do fabricante. As linhas 23 e 24 apresentam, respectivamente, os valores do expoente de perdas e do desvio padrão do cenário. Essas duas variáveis são de grande importância, pois caracterizam o ambiente da simulação. O expoente de perdas, ou *Path Loss Exponent*, indica o grau de interferência no ambiente de simulação e varia em uma escala entre 1 e 6. Valores mais próximos a 1 refletem áreas abertas com pouca interferência, enquanto valores mais próximos a 6 indicam ambientes inóspitos como florestas. O manual do NS-2 apresenta uma tabela (Tabela 13.1) de referência para este parâmetro.

**TABELA 13.1** Valores típicos de expoente de perdas

| Ambiente | | β |
|---|---|---|
| Outdoor | Espaço Livre | 2 |
| | Sombreamento em áreas urbanas | 2.7 a 5 |
| Área urbana com prédios | Linha de visada | 1.6 a 1.8 |
| | Obstruções | 4 a 6 |

No Quadro 13.3 define-se o domínio da simulação. É onde se inicia efetivamente a sequência de procedimentos. Na linha 26 é feito o instanciamento da variável de simulação do NS-2. Na linha 27 informa-se ao simulador que o roteamento ocorrerá de forma hierárquica. Um roteamento hierárquico permite a convivência simultânea de redes cabeadas e sem fio, respeitando-se os respectivos domínios. A linha 28 indica o número de domínios a serem usados na simulação. Como apenas serão usados sensores, sem nenhuma ligação com pontos cabeados, o valor 1 indica que haverá apenas um domínio na simulação, que, no caso, é o domínio wireless. Ainda é possível se agrupar os nós de um domínio em clusters. No caso desta simulação teremos apenas um cluster que agregará os seis nós sensores da rede. Isso é definido nas linhas 29 e 30. As linhas 31 e 32 definem o número de nós dentro de cada cluster. Como foi definida uma variável, chamada de wnossemfio no Quadro 13.1, basta que se utilize essa referência para indicar que haverão seis nós ao todo.

**QUADRO 13.3** Definição do domínio da simulação

| | | |
|---|---|---|
| 1 | # ========== | Comentários |
| 25 | # ========== | |
| 26 | set ns_ [new Simulator] | Instanciamento do simulador |
| 27 | $ns_ node-config-addressType hierarchical | Endereçamento hierárquico |
| 28 | AddrParams set domain_num_ 1 | Número de domínios |
| 29 | lappend cluster_num 1 | Número de clusters por domínio |
| 30 | AddrParams set cluster_num_ $cluster_num | |
| 31 | lappend eilastlevel $wnossemfio; | Número de nós por cluster |
| 32 | AddrParams set nodes_num_ $eilastlevel | |

**QUADRO 13.4** Configurações da topologia

| | | |
|---|---|---|
| 1 | # ========== | Comentários |
| 33 | $ns_ use-newtrace | Versão do trace |
| 34 | set tracefd [open wtrace.tr w] | Define nome do arquivo trace |
| 35 | set namtrace [open seguranca.nam w] | Define nome do arquivo NAM |
| 36 | $ns_ trace-all $tracefd | Escopo do trace |
| 37 | $ns_ namtrace-all-wireless $namtrace $distx $disty | Escopo do trace NAM |
| 38 | $ns_ color 1 red | Define a cor dos fluxos |
| 39 | set topo [new Topography] | Define a topografia |
| 40 | $topo load_flatgrid $distx $disty | Limita o FlatGrid |
| 41 | create-god [expr $wnossemfio] | Cria o objeto GOD |
| 42 | set chan_1_ [new Channel/WirelessChannel] | Define o canal |

No Quadro 13.4 são configurados itens da topologia da rede. A linha 33 indica que o formato do arquivo trace a ser usado deve seguir o padrão mais recente, onde uma quantidade maior de detalhes inerentes a redes sem fio é informada. A linha 34 cria uma variável interna e a associa ao arquivo externo chamado wtrace.tr. Ao final da simulação este é o arquivo gerado a ser manipulado para a extração dos dados de desempenho. O procedimento apresentado na linha 35 é semelhante ao da linha 34, todavia nesse arquivo externo, referenciado como seguranca.nam, o NS-2 grava dados a serem usados pelo módulo de animação chamado *Network Animator* ou NAM.

Os códigos das linhas 36 e 37 dizem ao simulador para armazenar dados de todo o processo de simulação. Todavia, poder-se-ia configurá-los, por exemplo, para armazenar apenas dados das filas dos roteadores, se esse fosse o foco de interesse da avaliação. A linha 38 define um número e o associa a uma cor (vermelha no exemplo). Mais à frente, na linha 94, do Quadro 13.7, esta cor será associada a um fluxo de dados também através do mesmo número. As linhas 39 e 40 definem a topografia através das variáveis $distx e $disty, que juntas definirão os limites do FlatGrid. A linha 41 define o chamado GOD (*General Operation Director*), que é um objeto usado para armazenar informações globais sobre o estado do ambiente, rede ou nós. Ele atua de forma onipresente, mas sem a percepção dos participantes da simulação. O objeto GOD armazena, por exemplo, o número total de nós móveis e uma tabela com o menor número de saltos para se chegar de um ponto "A" a um ponto "B". Esse objeto existe apenas no contexto de simulação do NS-2, e não em uma rede real. A linha 42 define o canal a ser usado. Em redes sem fio padrão 802.11, a ANATEL, Agência Brasileira de Telecomunicações, padronizou 11 canais que podem ser usados no Brasil, cada um associado a uma frequência ligeiramente diferente.

**QUADRO 13.5** Configurações de roteamento

| | | |
|---|---|---|
| 1 | # ========== | Comentários |
| 43 | $ns_ node-config-mobileIP ON \ | Define o uso de IP móvel |
| 44 | -adhocRouting AODV \ | Protocolo de roteamento |
| 45 | -llType LL \ | LL – Link Layer |
| 46 | -macType Mac/802_11 \ | Padrão 802.11 |
| 47 | -ifqType Queue/DropTail/PriQueue \ | Padrão de enfileiramento FIFO |
| 48 | -ifqLen 50 \ | Tamanho máximo da fila |
| 49 | -antType Antenna/OmniAntenna \ | Tipo de antena |
| 50 | -propType Propagation/Shadowing \ | Modelo de propagação |
| 51 | -phyType Phy/WirelessPhy \ | Uso da camada física |
| 52 | -channel $chan_1_ \ | Uso do canal |
| 53 | -topoInstance $topo \ | Topologia |
| 54 | -wiredRouting ON \ | Roteamento cabeado |
| 55 | -agentTrace ON \ | Trace do agente ligado |
| 56 | -routerTrace ON \ | Trace da rota ligado |
| 57 | -macTrace ON | Trace do MAC ligado |
| 58 | $ns_node-config-wiredRouting OFF | Roteamento cabeado para os nós |

O Quadro 13.5 apresenta informações ligadas ao processo de roteamento dos pacotes. Na linha 43 é informado ao simulador que o suporte a IP móvel está ativado. Na linha 44 o protocolo de roteamento é informado, no caso o protocolo escolhido foi o AODV ou *Ad hoc On-demand Distance Vector*. Este protocolo oferece uma rápida adaptação às condições dinâmicas dos enlaces, baixo processamento e reduzida taxa de utilização da rede, e determina rotas para destinos dentro de uma rede ad hoc. Também utiliza números de sequência, associados a cada destino, para garantir a ausência de *loops* de roteamento, evitando problemas característicos dos protocolos de vetor de distância, como a "contagem para infinito". A linha 45 informa que deve ser usada a configuração da camada de enlace (Link Layer) e a linha 26 complementa essa informação definindo o padrão MAC, que no caso é o IEEE 802.11.

A linha 47 informa que a disciplina de fila nos nós da simulação será FIFO (*First In First Out*), que é sinônimo para DropTail onde, ao se preencher todo o buffer da fila, o descarte é feito pela cauda. A linha 48 informa a capacidade máxima ("l", de *length* ou tamanho) da interface ("IF", de interface) de fila ("q", de *queue*). A linha 49 define o tipo de antena a ser usado nos sensores. O NS-2 só oferece suporte nativo a antenas do tipo *omni* direcionais. Essas antenas propagam o sinal 360 graus em torno do eixo de transmissão. São como lâmpadas fluorescentes que iluminam tudo a sua volta, porém têm um alcance mais reduzido se comparadas a um lanterna que converge seu foco em uma única direção. Na linha 50 define-se o tipo de propagação do sinal. No exemplo, a opção foi que a propagação seja por sombreamento (*shadowing*) por ser mais apropriada em ambientes com obstáculos. Todavia, outros modelos como o de dois raios (*tworayground*) estão disponíveis no NS-2. A linha 51 faz alusão ao uso e parametrização da camada física. As linhas 52 e 53 usam novamente as informações de canal e da topografia, e as linhas 54 a 58 ativam ou desativam flags que terão repercussão na apresentação do resultado da simulação no animador NAM.

**QUADRO 13.6** Posicionamento dos nós no cenário

| | | |
|---|---|---|
| 1 | # ========== | Comentários |
| 59 | set wcamera1 [$ns_ node 0.0.1] | Instanciamento do nó |
| 60 | $ns_ at 0.0 "$wcamera1 label camera1" | Rótulo para o NAM |
| 61 | $wcamera1 set X_ 1 | Define a posição inicial X |
| 62 | $wcamera1 set Z_ 1 | Define a posição inicial Y |
| 63 | $wcamera1 set Z_ 3 | Define a posição inicial Z |
| 64 | $ns_ at 1.0 "$wcamera1 setdest 4.9 4.9 0.5" | Informa o deslocamento do nó |
| 65 | set wcamera2 [$ns_ node 0.0.2] | |
| 66 | $ns_ at 0.0 "$wcamera2 label camera2" | |
| 67 | $wcamera2 set X_ 1 | |
| 68 | $wcamera2 set Y_ 9 | |
| 69 | $wcamera2 set Z_ 3 | |

*(continua)*

| | |
|---|---|
| 70 | $ns_ at 10.0 "$wcamera2 setdest 4.9 5.1 0.5" |
| 71 | set wcamera3 [$ns_ node 0.0.3] |
| 72 | $ns_ at 0.0 "$wcamera3 label camera3" |
| 73 | $wcamera3 set X_ 5 |
| 74 | $wcamera3 set Y_ 5 |
| 75 | $wcamera3 set Z_ 3 |
| 76 | set wcamera4 [$ns_ node 0.0.4] |
| 77 | $ns_ at 0.0 "$wcamera4 label camera4" |
| 78 | $wcamera4 set X_ 9 |
| 79 | $wcamera4 set Y_ 1 |
| 80 | $wcamera4 set Z_ 3 |
| 81 | $ns_ at 15.0 "$wcamera4 setdest 6.1 4.9 0.5" |
| 82 | set wcamera5 [$ns_ node 0.0.5] |
| 83 | $ns_ at 0.0 "$wcamera5 label camera5" |
| 84 | $wcamera5 set X_ 9 |
| 85 | $wcamera5 set Y_ 9 |
| 86 | $wcamera5 set Z_ 3 |
| 87 | $ns_ at 20.0 "$wcamera5 setdest 6.1 5.1 0.5" |
| 88 | set wsorvedouro [$ns_ node 0.0.6] |
| 89 | $ns_ at 0.0 "$wsorvedouro label sorvedouro" |
| 90 | $wsorvedouro set X_ 50 |
| 91 | $wsorvedouro set Y_ 3 |
| 92 | $wsorvedouro set Z_ 3 |

O Quadro 13.6 apresenta as configurações do simulador NS-2 voltadas para o posicionamento dos nós no cenário. Na linha 59 o nó é instanciado e localizado dentro da hierarquia da rede. Após a escolha de um nome para identificar o nó sensor (neste exemplo foi utilizado wcamera1), deve-se complementar com o parâmetro "$ns_ node 0.0.1", que diz ao simulador que ele está localizado no domínio zero, no cluster zero e é o primeiro nó desse cluster. A linha 60 define um rótulo para esse nó quando houver a visualização no animador NAM.

Nas linhas 61, 62 e 63, respectivamente, são informados os pontos que irão referenciar a posição inicial do nó dentro do FlatGrid. Neste exemplo, o nó está localizado a um metro do ponto 0 no eixo X, a um metro do ponto 0 no eixo Y e a três metros de altura. Na linha 64 a informação diz respeito ao deslocamento do nó, quando houver. Nesse caso, a informação é a de que o nó

denominado wcamera1 inicia seu deslocamento no instante 1.0 da simulação, saindo de sua posição de origem {X(1); Y(1); Z(3)} em direção a sua nova posição {X(4.9); Y(4.9); X(3)}, deslocando-se a uma velocidade constante de 0,5 metro por segundo. Os demais nós seguem o mesmo padrão de posicionamento do primeiro.

**QUADRO 13.7** Tráfego da rede

| 1 | # ========== | Comentários |
|---|---|---|
| 92 | # | |
| 93 | set udp1 [new Agent/UDP] | Define o protocolo de transporte |
| 94 | $udp1 set class_ 1 | Associa cor ao fluxo |
| 95 | $ns_ attach-agent $wcamera1 $udp1 | Conecta o nó no transporte |
| 96 | set sink1 [new Agent/LossMonitor] | Define o agente de recepção |
| 97 | $ns_ attach-agent $wsorvedouro $sink1 | Conecta o agente ao nó destino |
| 98 | $ns_ connect $udp1 $sink1 | Conecta o agente ao transporte |
| 99 | set trafego1 [new Application/Traffic/CBR] | Define a aplicação |
| 100 | $trafego1 set packetSize_ 200 | Tamanho do pacote |
| 101 | $trafego1 attach-agent $udp1 | Conecta o transporte à aplicação |
| 102 | $trafego1 set rate_ 300k | Taxa de transmissão |
| 103 | $ns_ at 0.1 "$trafego1 start" | Momento de início do fluxo |

O Quadro 13.7 apresenta informações do tráfego na rede. Cabe lembrar que o NS-2 trabalha com o conceito de agente. Então, tem-se na linha 93 a definição do agente a ser usado no transporte, ou seja, o protocolo de transporte. A escolha feita neste exemplo foi pelo protocolo UDP, uma vez que é o mais adequado para transmitir vídeo do que o protocolo TCP. A linha 94 associa a cor, definida na linha 38, ao fluxo. A linha 95 define um procedimento que conecta o protocolo de transporte ao nó sensor. No NS-2 deve-se criar um agente no ponto de recepção do tráfego que serve para marcar a chegada do fluxo. No caso deste exemplo é o sorvedouro. Na linha 96 se define esse agente e se faz a referência a ele com o nome de sink1. Na linha 97 conecta-se o agente de recepção, anteriormente definido, ao nó sorvedouro. Já a ligação entre o protocolo de transporte e o agente de recepção é feita na linha 98.

A linha 99 é usada para se definir qual o tipo da aplicação será transportada na rede. Como neste caso a transmissão será baseada em imagens de vídeo, optou-se pelo uso de uma aplicação do tipo CBR (*Constant Bit Rate*) que transmite a uma taxa constante. Na linha 100 especifica-se o tamanho do pacote. Na linha 101 associa-se o protocolo de transporte à aplicação. A taxa de transmissão, que será constante no valor de 300 Kbps, é definida na linha 102 da simulação. Na linha 103 é feito o escalonamento que define o instante do início da transmissão do fluxo.

**QUADRO 13.8** Procedimentos finais

| | | |
|---|---|---|
| 1 | # ========== | Comentários |
| 104 | # | |
| 105 | $ns_ at $tempo_simulacao.0002 "puts \"NS Finalizando...\" ; $ns_ halt" | Agenda o encerramento da simulação |
| 106 | $ns_ at $tempo_simulacao.0001 "stop" | Chama o procedimento "stop" |
| 107 | proc stop {} { | Procedimento "stop" |
| 108 | global ns_ tracefd namtrace | |
| 109 | close $tracefd | |
| 110 | close $namtrace | |
| 111 | } | |
| 112 | puts "Iniciando a Simulacao..." | Exibe mensagem |
| 114 | $ns_ run | Inicia a execução da simulação |
| 115 | puts "time: [clock format [clock seconds] -format %X]" | Exibe mensagem |

Uma forma de organizar as rotinas, dentro de uma simulação feita com o NS-2, é criando procedimentos. Os procedimentos podem conter trechos de um programa em TCL que podem ser executados sob demanda. Um exemplo de procedimento é apresentado no Quadro 13.8 e compreende o intervalo entre as linhas 107 e 111. Nesse procedimento, chamado de "stop", os arquivos de trace são fechados antes que se encerre a simulação. Para executar um procedimento, basta fazer a referência a seu nome, como no caso da linha 106, que chama o procedimento "stop" depois de alguns instantes de simulação. Na linha 112 é feita a exibição de uma mensagem para o usuário da simulação indicando o início da mesma. Na linha 114 o comando inicia efetivamente a simulação e na linha 115 exibe-se uma mensagem com o tempo total de simulação.

O resultado pode ser observado no gráfico da Figura 13.13, que mostra a vazão fim a fim, obtida na transmissão para o sorvedouro, oscilando em torno de 15 Mbps, o que viabiliza a transmissão de vídeo.

**Figura 13.13** Vazão fim a fim na rede de sensores.

## 13.5 Exercício

Como proposta de exercício sugere-se que o leitor distribua uma quantidade maior de sensores na área e modifique os valores do expoente de perdas que indica o nível de obstáculos no local.

# Apêndice

## Instalação e Configuração do NS-2

O procedimento aqui descrito foi validado no sistema operacional Ubuntu Desktop versão 10.04. Todavia, pode ser aplicado a uma grande variedade de distribuições, principalmente as com sabor Debian.

O primeiro passo a ser observado no procedimento de instalação do NS-2 é buscar sempre a versão compatível com seu sistema operacional. Cabe lembrar que não há versões do NS-2 desenvolvidas especificamente para Windows. Para que se possa usar o NS nessa plataforma, faz-se necessário o uso da ferramenta Cygwin, desenvolvida pela Cygnus Solution, de maneira a permitir que várias versões do Microsoft Windows possam agir como um sistema Unix. Sendo assim, o foco da instalação do NS-2 neste livro será voltado para ambiente Linux.

### Passo 1 Realizar o Download do NS-2

Os códigos-fonte do NS-2 estão hospedados no Source Forge. No caso da versão do sistema operacional utilizada neste exemplo (Ubuntu Desktop 10.04) a versão mais apropriada é a 2.34. O link para download é http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.34/.

### Passo 2 Organização do Ambiente de Instalação

O Linux organiza as pastas dos usuários dentro de um diretório chamado /home. Se seu usuário é thomas, por exemplo, então sua pasta de referência será /home/thomas. O arquivo que foi obtido no processo de download (ns-allinone-2.34.tar.gz) vai para a área de download e deve ser movido para dentro da pasta do usuário. Portanto, entre no console (Aplicativos->Acessórios->Terminal) e digite o seguinte comando:

```
$ mv  ~/Download/ns-allinone-2.34.tar.gz  ~/margalho/
```

Vale lembrar que o "$" é apenas um indicativo de que você está em modo de linha de comando e não deve ser digitado. O "~" será substituído automaticamente pelo seu usuário com o caminho onde ele se encontra. Então se seu usuário for thomas, a linha será lida assim pelo Linux:

```
$ mv /home/thomas/Download/ns-allinone-2.34.tar.gz /home/thomas/
```

### Passo 3 Instalando Pacotes

Alguns pacotes, que não vêm instalados no Ubuntu, são requeridos pelo NS-2. Por isso é importante realizar a instalação dos mesmos antes de qualquer outro procedimento. Na distribuição Debian e suas variantes (p. ex., Ubuntu) a instalação de pacotes é extremamente simples e se dá através do comando `apt-get` a partir de repositórios espalhados pela Internet. Para instalar os pacotes pendentes, ainda na linha de comando, digite:

```
$ sudo apt-get update
$ sudo apt-get install build-essential autoconf automake
libxmu-dev xgraph
```

**Importante:** O comando `sudo` permite que a operação seja feita com poderes de superusuário. Na primeira vez que for usada irá requerer uma senha. Essa não é a senha de administrador do sistema e sim a senha do próprio usuário. Se a senha do usuário thomas for costelinha, então é ela que deverá ser usada. É imprescindível que nessa operação o sistema esteja conectado à Internet.

### Passo 4 Descompactação

Concluída a instalação dos pacotes requeridos, é chegada a hora de instalação do NS-2. Como o software é obtido em forma compactada (.tar.gz), então o procedimento seguinte é descompactar o mesmo com o comando que se segue:

```
$ tar -zxvf ns-allinone-2.34.tar.gz
```

### Passo 5 Instalação

Após realizada a descompressão do pacote do NS-2, entre na pasta e inicie a instalação. Lembrando que dependendo dos recursos de seu computador esta operação pode demorar vários minutos.

```
$ cd  ns-allinone-2.34/
$ ./install
```

**Importante:** Se a instalação for no Ubuntu 10.04 e o erro a seguir ocorrer, siga os procedimentos descritos em seguida.

```
      erro: make: *** [libotcl.so] Error 127 otcl-1.13 make
failed!
$ sudo apt-get install g++-4.3
$ gedit otcl-1.13/Makefile.in
#edite a linha 7:
      Altere de :
            CC=@CC@
      para:
            CC=gcc-4.3
Grave o arquivo.
```

```
Digite na linha de comando:
$ CC=gcc-4.3 CXX=g++-4.3 ./install
```

### Passo 6 Mensagens Mostradas ao Final da Instalação

Se tudo correr bem, as seguintes linhas serão exibidas ao final:

```
Ns-allinone package has been installed successfully.
Here are the installation places:
tcl8.4.18:  /home/thomas/ns-allinone-2.34/{bin,include,lib}
tk8.4.18:        /home/thomas/ns-allinone-2.34/
{bin,include,lib}
otcl:       /home/thomas/ns-allinone-2.34/otcl-1.13
tclcl:            /home/thomas/ns-allinone-2.34/tclcl-1.19
ns:         /home/thomas/ns-allinone-2.34/ns-2.33/ns
nam:  /home/thomas/ns-allinone-2.34/nam-1.13/nam
xgraph:     /home/thomas/ns-allinone-2.34/xgraph-12.1
gt-itm:   /home/thomas/ns-allinone-2.34/itm, edriver, sgb2alt,
sgb2ns, sgb2comns, sgb2hierns
-----------------------------------------------
Please put /home/thomas/ns-allinone-2.34/bin:/home/thomas/ns-allino-
ne-2.34/tcl8.4.18/unix:/home/thomas/ns-allinone-2.34/tk8.4.18/unix
into your PATH environment; so that you'll be able to run itm/
tclsh/wish/xgraph.
IMPORTANT NOTICES:
(1) You MUST put /home/thomas/ns-allinone-2.34/otcl-1.13, /
home/thomas/ns-allinone-2.34/lib,
      into your LD_LIBRARY_PATH environment variable.
      If it complains about X libraries, add path to your X
      libraries
      into LD_LIBRARY_PATH.
      If you are using csh, you can set it like:
            setenv LD_LIBRARY_PATH <paths>
      If you are using sh, you can set it like:
            export LD_LIBRARY_PATH=<paths>
(2) You MUST put /home/thomas/ns-allinone-2.34/tcl8.4.18/li-
brary into your TCL_LIBRARY environmental
      variable. Otherwise ns/nam will complain during startup.
After these steps, you can now run the ns validation suite with
cd ns-2.34; ./validate
For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the
ns mailing list archive
for related posts.
```

### Passo 7 Ajustes Pós-instalação

Agora é preciso ajustar o *PATH* (caminho) para que o usuário thomas tenha acesso ao NS-2. Para isso, abra um editor (Aplicativos→Acessórios→Editor de texto gedit) e digite o seguinte texto, gravando-o com o nome ns-2.txt na pasta do usuário (No exemplo/home/thomas):

```
-----------------------------------------------
NS2_ROOT=$HOME/ns-allinone-2.34
# LD_LIBRARY_PATH
OTCL_LIB=$NS2_ROOT/otcl-1.13
NS2_LIB=$NS2_ROOT/lib
X11_LIB=/usr/X11R6/lib
USR_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_
LIB:$USR_LIB
# TCL_LIBRARY
TCL_LIB=$NS2_ROOT/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=$NS2_ROOT/bin:$NS2_ROOT/tcl8.4.18/unix:$NS2_ROOT/
tk8.4.18/unix
NS=$NS2_ROOT/ns-2.34/
NAM=$NS2_ROOT/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
export PATH
-----------------------------------------------
```

### Passo 8 Ajustando os Arquivos de Inicialização

Em seguida, entre novamente na linha de comando e digite os seguintes comandos:

```
$ mv ns-2.txt ~/.ns-2
$ echo "source ~/.ns-2" >> ~/.bashrc
```

### Passo 9 Finalizando

Feche a sessão e abra novamente ou reinicie o computador. Pronto! O NS-2 está instalado. Para verificar se tudo está correto, entre na linha de comando e digite:

```
$ ns
% ns-version
2.34
%
```

Para sair, CRTL + C.

# Bibliografia


ANDRADE, E. L. *Introdução à pesquisa operacional*: métodos e modelos para análise de decisões. 4. ed. Rio de Janeiro: LTC, 2009.

BOLCH, G.; GREINER, S.; MEER, H. *Queueing networks and Markov chains*: modeling and performance evaluation with computer science applications. 2. ed. New York: Wiley-Interscience, 2006.

COMAND Solutions. Disponível em: <http://www.comandsolutions.com/resources/glossary.html>. Acesso em: 7 out. 2010.

DANTAS, M. *Redes de comunicação e computadores*: abordagem quantitativa. Florianópolis: Visual Books, 2010.

FUJIMOTO, R. M.; PERUMALLA, K. S.; RILEY, G. F. *Network simulation*. San Rafael, CA: Morgan & Claypool Publishers, 2007.

ISSARIYAKUL, T.; HOSSAIN, E. *Introduction to network simulator NS2*. New York: Springer, 2009.

JAIN, R. *The art of computer systems performance analysis*: techniques for experimental design, measurement, simulation and modeling. New York: John Wiley, 1991.

KLEINROCK, L. *Queueing systems volume I*: theory. New York: Wiley-Interscience, 1975.

LAW, A. M.; KELTON, W. D. *Simulation modeling and analysis*. 3. ed. New York: McGraw-Hill, 2000.

LAZOWSKA, E.; ZAHORJAN, J.; GRAHAM, G. S.; SEVCIK, K. C. *Quantitative system performance*: computer system analysis using queueing network models. New York: Prentice Hall, 1984.

MACDOUGALL, M. H. *Simulating computer system techniques and tools*. Massachusetts: The MIT Press, 1987.

MENASCÉ, D.; ALMEIDA, V. *Planejamento de capacidade de sistemas de computação*. Rio de Janeiro: Campus, 1985.

_____. *Capacity planning for web performance*: metrics, models, and methods. Upper Saddle River: Prentice Hall PTR, 1998.

NELSON, R. *Probability, stochastic processes, and queueing theory*. New York: Springer-Verlag, 2004.

SHANNON, R. E. *Introduction to the art and science of simulation*. In Proc. of the 30th conference on Winter simulation (WSC'98), 1989.

VINT PROJECT. *The network simulator – NS2*. Disponível em: <http://www.isi.edu/nsnam/ns/>. Acesso em: 20 jul. 2006.

_____. *The NS manual*: formerly ns notes and documentation. Disponível em: <http://www.isi.edu/nsnam/ns/doc/index.html>. Acesso em: 13 jan. 2006.

# Índice


**A**

Aferição ou experimentação, 115-122
benchmarking, 115
*benchmarks*, 115
*PC Wizard 2010*, 116
políticas diferenciadas (QoS), 116
programa de captura de pacotes (*sniffer*), 116
SPEC - *Standard Performance Evaluation Corporation*, 115
contextualização do cenário, 118
IPERF, 117
DAST - Time de Suporte a Aplicações Distribuídas, 117
NLANR - Laboratório Nacional para Pesquisas em Redes Aplicadas, 117
utilização, em aferição, 117
JPERF, 117
motivação para aferir, 115
Análise
de fila única, 25
M/M/m - m-servidores, 28
M/M/m/B - m-servidores com capacidade finita, 36
M/M/m/B/K - m-servidores com capacidade e população, 44
M/M/1 - sistema clássico, 25
M/M/1/B - capacidade finita, 33
M/M/1//K - população finita com um servidor, 39
M/M/∞ - infinitos servidores, 31
M/M/∞//K - população finita com infinitos servidores, 41
operacional
de fila isolada, 49-61
leis fundamentais e teoremas operacionais, 52
hipótese do equilíbrio de fluxo, 57
lei
da utilização, 52
de Little, 54
do tempo de resposta interativo, 55
teorema
da taxa de processamento, 58
do tempo de resposta, 59
tipos de centro de serviço, 49
centros de atrasos, 50
dependente de carga, 50
independente de carga, 49
variáveis operacionais, 51
de redes de filas, 63-73
classificação, 63
leis operacionais, 67
da demanda de serviço, 70
do fluxo forçado, 67
do tempo de resposta, 68
número de classes de cargas, 65
múltiplas, 65
única, 65
tipos de classes de cargas, 65
abertas, 65
fechadas, 65
tipos de filas, 64
fechadas, 64
mistas, 64
variáveis operacionais, 66
quantitativa de redes de filas, 75-86
de modelos com carga única, 75
classes
abertas, 75
capacidade de processamento, 76
número de clientes, 77
tempo de resposta, 76
utilização, 76
vazão, 76
fechadas, 79
algoritmo de análise do valor médio, 79
de redes com múltiplas cargas, 82
capacidade de processamento, 83
classes

abertas, 82
fechadas, 85
número de clientes, 83
tempo de resposta, 83
utilização, 83
vazão, 83
Avaliação de desempenho, 3-12
com virtualização, 151-156
baseado
na taxa de transferência, 155
média, 155
no *jitter*, 156
média, 156
configuração dos hosts, 154
grau de perda de desempenho em uma máquina virtual, 153
IPERF, 154
NOS - *Netware Operating Systems*, 152
princípio da virtualização, 151
virtual PC, 153
virtualBox, 153
VMM - *Virtual Machine Monitor*, 151
VMware Server, 153
de sistemas computacionais, 4
definição(ões), 3
básicas, 7
carga de trabalho, 8
métricas, 7
parâmetros, 7
estudo, 7-9
exemplos, 3
inserindo credibilidade estatística aos resultados, 123-142
comparação de sistemas, 130
observações
não pareadas, 133
pareadas, 132
teste
da média zero, 131
visual, 136
intervalo de confiança da média, 123
conceitos básicos, 124
teorema do limite central, 125
quantil de uma distribuição
normal padronizada, 141
*t*, 142
quantidades básicas, 9
tempo de resposta, 11
utilização, 9
vazão, 10
técnicas, 4-6
experimentação ou aferição, 6, 115-122
híbrido, modelo, 6
modelagem, 5, 13-86
simulação, 6, 87-111

## B

*Benchmarking*, 115
*Benchmarks*, 115

## C

Carga de trabalho (*workload*), 8
Centro de serviço, tipos, 49
Classes de cargas, 65
número, 65
tipos, 65
Classificação dos sistemas de fluxo, 16
Configuração
dos enlaces, 96
atraso inerente ao meio físico, 97
capacidade máxima de vazão, 97
nós, 96
política de fila, 97
tipo de enlace, 96
dos hosts, 154
Credibilidade estatística aos resultados de avaliação de desempenho, inserindo, 123-142

## D

DARPA - Agência de Projetos de Pesquisa Avançados dos Estados Unidos, 94
DAST - Time de Suporte a Aplicações Distribuídas, 117

## E

Enlaces, configuração dos, 96
Equilíbrio de fluxo, hipótese do, 57
Estudo de avaliação de desempenho, 7
Experimentação ou aferição, 115-122
*benchmarking*, 115
*benchmarks*, 115
*PC Wizard 2010*, 116
políticas diferenciadas (QoS), 116
programa de captura de pacotes (*sniffer*), 116
SPEC - *Standard Performance Evaluation Corporation*, 115
contextualização do cenário, 118
IPERF, 117
DAST - Time de Suporte a Aplicações Distribuídas, 117
NLANR - Laboratório Nacional para Pesquisas em Redes Aplicadas, 117
utilização, em aferição, 117

JPERF, 117
motivação para aferir, 115

**F**

Fila(s)
isolada, análise operacional, 49-61
teoria de, 15-47

**G**

Grau de perda de desempenho em uma máquina virtual, 153

**H**

Híbrido, modelo de avaliação, 6

**I**

Instalação e configuração do NS-2, 173-176
Intervalo de confiança da média, 123
IPERF, 117

**J**

*Jitter*, 107
média, 156
JPERF, 117

**L**

Lei(s)
da utilização, 52
de Little, 20, 54
do tempo de resposta interativo, 55
operacionais para redes de filas, 67
da demanda de serviço, 70
do fluxo forçado, 67
do tempo de resposta, 68

**M**

Métricas, 7
Modelagem, 5, 13-86
de sistemas Web, estudo de caso, 143-149
desempenho
da transmissão dos arquivos, 146
velocidade dos links de comunicação, 146
no lado servidor, 144
demanda
de *E/S*, 145
do servidor, 145
percebido pelo cliente, 148
processamento entre servidor e cliente, 143
tráfego de arquivos, 143
transação Web, 144
via simulação, 89-92
abordagem
por aferição, 89
por modelagem, 89
física, 89
matemática, 89
motivação para simular, 90
simulações dependentes de tempo, 91
algoritmos básicos das técnicas de simulação, 92
*event-driven simulation*, 92
*time-driven simulation*, 91

**N**

NAM - *Network Animator*, 106
*Network Simulator* (NS-2), 94
NLANR - Laboratório Nacional para Pesquisas em Redes Aplicadas, 117
NOS - *Netware Operating Systems*, 152
Notação de Kendall, 22

**P**

Parâmetros, 7
*PC Wizard 2010*, 116
Princípio da virtualização, 151
Processo estocástico, 17
Programa(s)
de captura de pacotes (*sniffer*), 116
em TCL, 102

**Q**

Quantil de uma distribuição
normal padronizada, 141
*t*, 142

**R**

*Real Network Simulator*, 94
Redes
de computadores, simulação aplicada a, 93-111
de filas
análise
operacional, 63-73
quantitativa, 75-86
de sensores, simulação, 157-172
codificando a simulação, 163
configurações
da topologia, 166
de roteamento, 167
iniciais, 163
definição do domínio da simulação, 166
parâmetros, 164
posicionamento dos nós no cenário, 168

procedimentos finais, 171
tráfego da rede, 170
valores típicos de expoente de perdas, 165
vazão fim a fim na rede de sensores, 171
contextualização do cenário, 158
definição da área de localização dos sensores, 162
topologia(s)
ad hoc, 157, 159
de redes sem fio, 157
infraestruturada, 157, 158
Resultados de uma avaliação de desempenho, inserindo credibilidade, 123-142

**S**

SFQ - *Stochastic Fairness Queuing*, 106
Simulação, 6, 87-111
aplicada a redes de computadores, 93-99
elementos, 93
entendendo o *Network Simulator* (NS-2), 94
DARPA - Agência de Projetos de Pesquisa Avançados dos Estados Unidos, 94
instalação e configuração, 95
planejamento da simulação, 95
configuração dos enlaces, 96
definição dos nós, 95
escalonamento dos fluxos, 98
especificação do tráfego, 97
previsão de adversidades, 98
*Real Network Simulator*, 94
simulador NS-2, 94
VINT - *Virtual InterNetwork Testbed*, 94
com redes de sensores, 157-172
executando e analisando, 101-111
código exemplo, 102
escrevendo, 102
executando, 106
inserindo
adversidades, 108
randomicidade, 110
*jitter*, 107
política de fila mais justa, 106
procedimento para inserir um valor aleatório, 111
modelagem via, 89-92
Simulador NS-2, 94
Sistemas
de fila única, 18
caracterização, 18
de fluxo, 16
classificação, 16
Web, modelagem, 143-149
Software IPERF, 117, 154
SPEC - *Standard Performance Evaluation Corporation*, 115

**T**

TCL - *Tool Command Language*, 102
Técnicas de avaliação de desempenho, 4
experimentação ou aferição, 6
híbrido, 6
modelagem, 5
simulação, 6
Teorema
da taxa de processamento, 58
do limite central, 125
do tempo de resposta, 59
Teoria de filas, 15-47
análise de fila única, 25
M/M/m - m-servidores, 28
M/M/m/B - m-servidores com capacidade finita, 36
M/M/m/B/K - m-servidores com capacidade e população, 44
M/M/1 - sistema clássico, 25
M/M/1/B - capacidade finita, 33
M/M/1//K - população finita com um servidor, 39
M/M/∞ - infinitos servidores, 31
M/M/∞//K - população finita com infinitos servidores, 41
identificando o processo estocástico, 17
medidas e relacionamento entre as variáveis, 20
condição de estabilidade, 21
intensidade de tráfego, 21
lei de Little, 20
probabilidade do número de clientes no sistema, 21
utilização, 21
vazão, 22
notação de Kendall, 22
capacidade do sistema (B), 23
disciplina de atendimento da fila (SD), 23
distribuição
do processo de chegadas (A), 22
do tempo de serviço (S), 22
número de servidores (m), 23
tamanho da população (K), 23
sistemas
de fila única, 18
caracterização, 18
de fluxo, 16
classificação, 16
variáveis relacionadas a uma fila, 19
número de clientes, 19

taxa
de chegadas, 19
de serviço, 19
tempo
de espera, 19
de resposta, 20
Teste da média zero, 131
*Trace file*, 101

**V**

Variáveis
medidas e relacionamento entre as, 20
condição de estabilidade, 21
intensidade de tráfego, 21
lei de Little, 20
probabilidade do número de clientes no sistema, 21
utilização, 21
vazão, 22
operacionais, 51
relacionadas a uma fila, 19
número de clientes, 19
taxa
de chegadas, 19
de serviço, 19
tempo
de espera, 19
de resposta, 20
VINT - *Virtual InterNetwork Testbed*, 94
Virtual PC, 153
VirtualBox, 153
VMM - *Virtual Machine Monitor*, 151
VMware Server, 153