
Provisão integrada de QoS relativa e absoluta em
serviços computacionais interativos com requisitos
de responsividade de tempo real

Priscila Tiemi Maeda Saito

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura : _____

**Provisão integrada de QoS relativa e absoluta em
serviços computacionais interativos com requisitos
de responsividade de tempo real¹**

Priscila Tiemi Maeda Saito

Orientador:

Prof. Dr. Francisco José Monaco

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para a obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Fevereiro/2010

¹Este trabalho tem o apoio financeiro do **CNPq** (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

Agradecimentos

Primeiramente a Deus por ter me dado forças e competência para a realização deste trabalho.

Aos meus pais Kiokuni Saito e Yokie Maeda Saito, minha irmã Larissa Midori Maeda Saito e toda a minha família, por todo Amor, carinho e compreensão e por sempre estarem ao meu lado, me apoiando e contribuindo para a minha formação pessoal e profissional.

Ao meu namorado Pedro Henrique Bugatti pelo companheirismo, por sempre estar ao meu lado, me apoiando e me incentivando a enfrentar os desafios e a superar os obstáculos. Obrigada pelos conselhos, por todo o seu carinho e Amor, que me ajudaram a renovar as energias e a continuar o trabalho.

Ao meu orientador Prof. Dr. Francisco José Monaco por ter me concedido a oportunidade de realizar o presente projeto, por sua orientação e pela confiança em meu trabalho.

À Profa Dra Kalinka Regina Lucas Jaquie Castelo Branco pelo apoio e pela disposição constante em ajudar. Obrigada pela oportunidade e honra de ter trabalhado e continuar trabalhando ao seu lado.

A todos os amigos do Grupo de Sistemas Distribuídos e Programação Concorrente (GSDPC), companheiros de laboratório, que de alguma maneira contribuíram para o desenvolvimento desse trabalho.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo apoio financeiro.

E por fim a todos que de forma direta e indireta contribuíram para a produção deste trabalho.

Muito Obrigada!

Resumo

Aplicações de sistemas computacionais emergentes atribuindo requisitos de resposta na forma de tempo de resposta requerem uma abordagem de sistemas de tempo real. Nesses sistemas, a qualidade de serviço é expressa como garantia das restrições temporais. Um amplo leque de técnicas para provisão de QoS encontram-se na literatura. Estas técnicas são baseadas tanto na diferenciação de serviço (QoS relativa), quanto na especificação de garantia de desempenho (QoS absoluta). Porém, a integração de QoS relativa e absoluta em nível de aplicação não tem sido tão explorada. Este trabalho realiza o estudo, a análise e a proposta de um método de escalonamento de tempo real em um ambiente simulado, baseado em contratos virtuais adaptativos e modelo re-alimentado. O objetivo é relaxar as restrições temporais dos usuários menos exigentes e priorizar usuários mais exigentes, sem degradar a qualidade do sistema como um todo. Para tanto, estratégias são exploradas em nível de escalonamento para o cumprimento dos contratos especificados por requisitos de tempo médio de resposta. Os resultados alcançados com o emprego do método proposto sinalizam uma melhoria em termos de qualidade de serviço relativa e absoluta e uma melhor satisfação dos usuários. Este trabalho também propõe uma extensão para os modelos convencionalmente estudados nesse contexto, ampliando a formulação original de duas classes para n classes de serviços.

Abstract

Emerging computer system application posing responsiveness requirement in the form of response time demand a real-time system approach. In these systems, the quality of service is expressed as guarantees on time constraints. A wide range of techniques for QoS provision is found in the literature. These techniques are based both on either service differentiation (relative QoS) or specification of performance guarantees (absolute QoS). However, integrated provision of both relative and absolute QoS at application level is not as well explored. This work conducts the study, analysis and proposal of a real time scheduling method in a simulated environment. This method is based on adaptive virtual contracts and feedback model. The goal is to relax the time constraints of less demanding users and prioritize the time constraints of most demanding users, without degrading the quality of the system as a whole. Strategies toward this goal are exploited in the system scheduling level and are aimed at the problem of fulfilling service-level agreements specifying average response times requirements. The results achieved with the proposed method indicate an improvement in relative and absolute QoS and a better user satisfaction. This work also proposes an extension to the conventional models studied in this context, extending the original formulation of two classes for n classes of services.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas e Siglas	xiii
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Motivação	2
1.3 Objetivos	4
1.4 Organização do Documento	4
2 Qualidade de Serviço	7
2.1 Considerações Iniciais	7
2.2 QoS em Nível de Rede	8
2.2.1 Serviços Integrados	8
2.2.2 Serviços Diferenciados	9
2.3 QoS em Nível de Aplicação	10
2.3.1 QoS Relativa	11
2.3.2 QoS Absoluta	13
2.4 Considerações Finais	16
3 Sistemas de Tempo Real	17
3.1 Considerações Iniciais	17
3.2 Sistemas de Tempo Real	17
3.2.1 Sistemas <i>Hard-RT</i> , <i>Soft-RT</i> e <i>Firm-RT</i>	18
3.2.2 Restrições Temporais	19
3.2.3 Caracterização das Tarefas	20

3.3	Escalonamento em Sistemas RT	21
3.3.1	Classificação do Escalonamento	21
3.3.2	Algoritmos de Escalonamento	22
3.4	Considerações Finais	30
4	Simulação de Sistemas	31
4.1	Considerações Iniciais	31
4.2	Simulação de Sistemas	32
4.2.1	Linguagens de Simulação	33
4.2.2	Simulação de Filas	33
4.3	Considerações Finais	38
5	Integrando QoS Relativa e Absoluta	39
5.1	Considerações Iniciais	39
5.1.1	Conceitos Relacionados	40
5.2	Método Proposto	40
5.2.1	Metodologia	43
5.3	Experimentos	44
5.3.1	Descrição dos Cenários	44
5.4	Resultados	45
5.4.1	Análise do Comportamento do Escalonador	46
5.4.2	Análise Quanto à Satisfação do Usuário	52
5.5	Considerações Finais	56
6	Proposta de um Novo Modelo Composto por n Classes de Serviços	59
6.1	Considerações Iniciais	59
6.2	Abordagem Proposta	60
6.2.1	Descrição dos Cenários	60
6.2.2	Resultados	62
6.3	Considerações Finais	65
7	Conclusões	67
7.1	Considerações Gerais	67
7.2	Principais Contribuições	69
7.3	Propostas para Trabalhos Futuros	71
	Referências Bibliográficas	73

Lista de Figuras

1.1	Representação do cenário apresentado pela política EBS: cumprimento dos contratos A e B em termos de QoS absoluta e ocorrência de inversão de prioridade em termos de QoS relativa.	3
2.1	Ilustração do Funcionamento de Sinalização do <i>RSVP</i> . Adaptada de Zhao et al. (2000)	9
3.1	Ilustração das Restrições Temporais. Adaptada de Leite (2007).	20
3.2	Caracterização das tarefas: (a) periódica; (b) esporádica e (c) aperiódica. Adaptada de Leite (2007).	20
3.3	Exemplo da Utilização do Algoritmo de Escalonamento FIFO. Adaptada de Gomes (2009).	23
3.4	Exemplo da Utilização do Algoritmo de Escalonamento SJF. Adaptada de Gomes (2009).	23
3.5	Exemplo da Utilização do Algoritmo de Escalonamento EDF. Adaptada de Peixoto (2008).	24
3.6	Representação do Modelo de Servidor <i>Web</i> Seqüencial com QoS. Adaptada de Casagrande (2007).	25
3.7	Representação do cenário de 10% de requisições A e 90% de requisições B, apresentados pelas políticas EBS e FIFO:	29
3.8	Representação do cenário de 50% de requisições A e 50% de requisições B, apresentados pelas políticas EBS e FIFO:	29
3.9	Representação do cenário de 90% de requisições A e 10% de requisições B, apresentados pelas políticas EBS e FIFO:	30
4.1	Ilustração dos Elementos e das Características de um Sistema de Fila. Adaptada de Prado (2004).	34

4.2	Exemplos de Centros de Serviços: (a) com 1 fila e 1 servidor; (b) com 2 filas e 1 servidor e (c) com 1 fila e 2 servidores. Adaptada de Soares (1992).	36
4.3	Ilustração dos Modelos de Redes de Filas: (a) Aberto; (b) Fechado e (c) Misto. Adaptada de Lazowska et al. (1984).	36
5.1	Representação do cenário de ocorrência de inversão de prioridade em termos de QoS relativa e solução considerando-se os contratos virtuais vc_A e vc_B .	41
5.2	Passos do método proposto.	43
5.3	Comparativo entre a política EBS e o método proposto. Cenários com 5% de variação contratual e proporção de requisições:	47
5.4	Comparativo entre a política EBS e o método proposto. Cenários com 5% de variação contratual e proporção de requisições:	48
5.5	Comparativo entre a política EBS e o método proposto. Cenários com 15% de variação contratual e proporção de requisições:	49
5.6	Comparativo entre a política EBS e o método proposto. Cenários com 15% de variação contratual e proporção de requisições:	50
5.7	Comparativo entre a política EBS e o método proposto. Cenários com 25% de variação contratual e proporção de requisições:	51
5.8	Comparativo entre a política EBS e o método proposto. Cenários com 25% de variação contratual e proporção de requisições:	52
5.9	Gráfico representando a Satisfação Contratual dos usuários em termos de:	53
5.10	Gráfico representando a Satisfação Contratual dos usuários em termos de:	54
5.11	Gráfico representando a Satisfação Contratual dos usuários em termos de:	54
5.12	Gráfico representando a Satisfação Contratual dos usuários em termos de:	55
5.13	Gráfico representando os ARTs abaixo da média contratada. Cenários com:	56
5.14	Gráfico representando os ARTs abaixo da média contratada. Cenários com:	56
6.1	Representação do cenário de 10% de requisições A, 45% de requisições B e 45% de requisições C, apresentados pela abordagem proposta ($nEBS$) e pela política FIFO:	63
6.2	Representação do cenário de 34% de requisições A, 33% de requisições B e 33% de requisições C, apresentados pela abordagem proposta ($nEBS$) e pela política FIFO:	64
6.3	Representação do cenário de 90% de requisições A, 5% de requisições B e 5% de requisições C, apresentados pela abordagem proposta ($nEBS$) e pela política FIFO:	65

Lista de Tabelas

4.1	Notação de Kendall	36
4.2	Descrição das Distribuições de Probabilidade	37
4.3	Notação e Definições Básicas dos Componentes de Redes de Filas.	37

Lista de Abreviaturas e Siglas

ANR	Algoritmo de Negociação Rigoroso.
ANC	Algoritmo de Negociação com o Cliente.
AMP	<i>Asymmetric Multicore Platforms.</i>
ART	<i>Average Response Time.</i>
ASPOL	<i>A Simulation Process - Oriented Language.</i>
CPU	<i>Central Processing Unit.</i>
CSIM	<i>C-based process-oriented simulation language.</i>
DiffServ	<i>Differentiated Services.</i>
DSField	<i>Differentiated Service Field.</i>
DSQ	<i>Dynamic Single Queue.</i>
EBS	<i>Exigency-Based Scheduling.</i>
EDF	<i>Earliest Deadline First.</i>
FCFS	<i>First Come First Served.</i>
FIFO	<i>First In First Out.</i>
GSDPC	Grupo de Sistemas Distribuídos e Programação Concorrente.
ICMC-USP	Instituto de Ciências Matemáticas e de Computação - USP.
IETF	<i>Internet Engineering Task Force.</i>
IntServ	<i>Integrated Services.</i>
IP	<i>Internet Protocol.</i>
ISO	<i>International Organization for Standardization.</i>
ISP	<i>Internet Service Provider.</i>
LaSDPC	Laboratório de Sistemas Distribuídos e Programação Concorrente.
LCFS	<i>Last Come First Served.</i>
MQ	<i>Multiple Queue.</i>
OSI	<i>Open System Interconnection.</i>
PRIAdap	Prioridades Adaptativo.

QoS	Qualidade de Serviço.
RT	<i>Real-Time.</i>
RSVP	<i>Resource Reservation Protocol.</i>
RSVAdap	Reserva Adaptativa de Recursos.
SFD	<i>Short Flow Differentiating.</i>
SIMULA	<i>SIMUlation LAnguage.</i>
SJF	<i>Shortest Job First.</i>
SLA	<i>Service Level Agreement.</i>
SMPL	<i>SiMulation Program Language.</i>
SQ	<i>Single Queue.</i>
SWDS	Servidor <i>Web</i> com Diferenciação de Serviços.
TCP	<i>Transmission Control Protocol.</i>
TOS	<i>Type of Service.</i>
USP	Universidade de São Paulo.
WFQ	<i>Weighted Fair Queuing.</i>

Introdução

1.1 Considerações Iniciais

A medida que o uso de sistemas computacionais interativos vêm se disseminando na realização de serviços presentes no cotidiano da sociedade, aplicações com requisitos temporais de responsividade tornam-se cada vez mais comuns. Sob essa perspectiva, exemplos como ensino a distância, telemedicina, comércio eletrônico, dentre outros, demandam abordagens de análise e síntese pertinentes ao domínio dos sistemas de tempo real - RT (*Real-Time*).

Algumas aplicações RT apresentam restrições mais rigorosas que outras. Em aplicações de controle e automação industrial, por exemplo, atrasos no tempo de atuação acima dos valores permitidos costumam implicar em falha grave. Já em serviços computacionais interativos, em geral, a ausência de garantias quanto ao cumprimento dos requisitos de desempenho e confiabilidade é um fator relevante, porém não é crítico. Em aplicações de comércio eletrônico, descartes e atrasos nos tempos de resposta podem causar perdas significativas de rendimentos devido à insatisfação dos usuários com o serviço recebido, no entanto, não trará sérios danos, apenas uma degradação de desempenho.

Sistemas projetados para atender esses tipos de aplicações são designados: sistemas *Hard-RT* (sistemas de tempo real mais rigorosos), *Firm-RT* e *Soft-RT* (sistemas de tempo real menos rigorosos) (ver Seção 3.2.1).

Quando a violação de restrições temporais de responsividade não implica diretamente em falha, mas em degradação do serviço, o cumprimento de requisitos de tempo real pode ser associado ao conceito de qualidade de serviço (QoS), referindo-se à capacidade

dos elementos de um sistema em prover garantias acerca de determinados parâmetros associados à percepção da qualidade de um dado serviço oferecido. Exige-se que tais parâmetros permaneçam dentro de limites bem definidos (Casagrande, 2007). No contexto de transmissão digital de dados, por exemplo, onde o conceito tem recebido substancial importância, a QoS pode relacionar-se às garantias quanto ao atraso, vazão ou perdas de unidades de dado, dentre outros parâmetros.

A especificação da qualidade de serviço pode ser realizada segundo duas abordagens distintas: relativa ou absoluta. Quando em termos relativos, compara-se o atendimento oferecido às diversas classes de serviço. O que se pretende garantir nesses casos é que uma classe de maior prioridade (valor de contrato menor) tenha um atendimento melhor (preferencial) em relação à uma classe com menor prioridade (valor de contrato maior). No atendimento baseado em QoS absoluta, por outro lado, pretere-se a prioridade como parâmetro de diferenciação entre classes em favor do estabelecimento de metas a serem garantidas para cada classe individualmente, de forma a garantir uma taxa mínima de serviço ou um atraso máximo de atendimento das requisições.

Em se tratando de escalonamento, a distinção essencial entre QoS relativa e absoluta é a de que enquanto a primeira tem a prioridade de atendimento como base do contrato de serviço a ser garantida pelo provedor, a segunda especifica metas e valores de desempenho a serem respeitadas, sendo a prioridade, neste caso, decidida dinamicamente em função do estado do sistema e das condições efetivas dos contratos estabelecidos.

Substancial contribuição para as técnicas de provisão de QoS é oriunda da área de redes de comunicação de dados, onde o desenvolvimento no nível de rede, tendo como referência o modelo OSI (*Open System Interconnection*), se destaca como foco de pesquisa. Algoritmos de roteamento para tráfego em tempo real são especialmente relevantes nesse campo. Por outro lado, iniciativas para a provisão de QoS em nível de aplicação começam a ganhar interesse (Kang et al., 2003; Teixeira et al., 2005; Wei & Xu, 2005). Diversas técnicas têm sido propostas a fim de viabilizar e evoluir as formas de disponibilização de QoS em nível de aplicação (Casagrande et al., 2007; Pan et al., 2008; Teixeira et al., 2004). Um tópico importante refere-se às políticas de escalonamento aplicadas ao atendimento de requisições pendentes no provedor de serviços.

1.2 Motivação

Por mais de uma década, pesquisadores têm explorado a área de qualidade de serviço. Entretanto, a maioria dos trabalhos encontrados preocupa-se com a proposta de novas políticas de escalonamento baseadas na diferenciação de serviço ou na especificação do cumprimento de requisitos temporais. No entanto, a integração de QoS relativa e absoluta não tem sido explorada da mesma forma.

A política EBS (*Exigency-Based Scheduling*) (Casagrande et al., 2007), dentro da linha

de pesquisa em escalonamento de tempo real junto ao grupo de Sistemas Distribuídos e Programação Concorrente, GSDPC-ICMC-USP, oferece como contribuição para a área uma técnica de escalonamento para sistemas computacionais interativos cujo desempenho no atendimento às especificações estocásticas de responsividade temporal se mostraram superiores às alternativas convencionais.

Dentre outras características (ver Seção 3.3.2), a EBS produz um balanceamento das cargas computacionais entre as classes de serviço A , cujo contrato é mais estrito (valor de contrato menor) e B , cujo contrato é mais relaxado (valor de contrato maior), de modo a poupar recursos em termos de demanda de potência destinados à classe B , para garantir recursos às classes que deles mais necessitam (classe A). Essa propriedade permite à EBS cumprir os contratos de QoS absoluta baseados em tempo médio de resposta - ART (*Average Response Time*) que outras técnicas convencionalmente empregadas em escalonamento de serviços interativos, como na *Web*, por exemplo.

Todavia, em determinados cenários de carga, para efetuar o balanceamento de potência computacional, a aplicação da EBS tem como efeito o fato de que uma classe, mesmo apresentando um maior valor de contrato, recebe um atendimento melhor (ART menor) em relação às demais classes. Em contratos de QoS absoluta, em que não existe qualquer relação de prioridade entre classes, esse fato não constitui um problema. Nesses casos, a relação dos tempos de resposta entre as classes de serviço pode ser qualquer, podendo inclusive inverter-se ao longo do tempo, desde que os limites superiores (C_A e C_B) estabelecidos para cada uma das classes sejam respeitados (Figura 1.1).

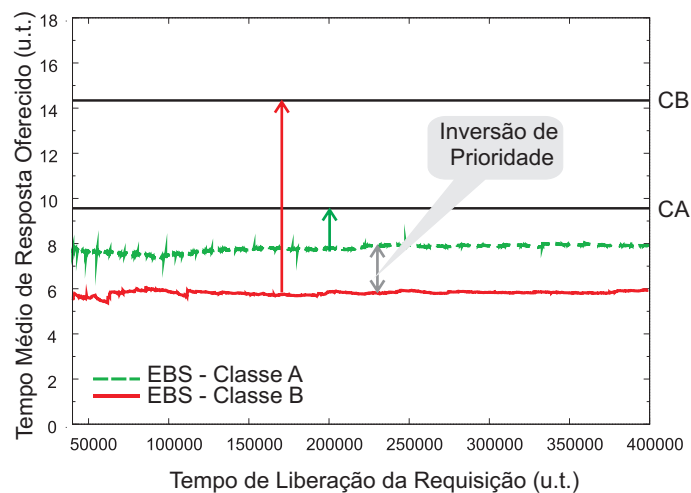


Figura 1.1: Representação do cenário apresentado pela política EBS: cumprimento dos contratos A e B em termos de QoS absoluta e ocorrência de inversão de prioridade em termos de QoS relativa.

Enquanto isso não seja um problema para o caso de QoS absoluta, tal fato passa a não ser atrativo para usuários que gastam mais com planos que definem menores ARTs contratuais. Mesmo tendo seus contratos específicos garantidos, o fato de em certos

momentos os usuários de classe *A* pagarem mais e receberem uma qualidade relativa inferior a outros que pagarem menos, pode ser percebido negativamente.

É plausível considerar qual a percepção dos usuários familiarizados ao modelo de atendimento preferencial da QoS relativa. Se uma classe com contrato mais estrito demanda mais recursos do sistema, e por isso deva arcar com custos maiores, prover-lhe um tempo de resposta superior (Figura 1.1) pode se lhe apresentar intuitivamente como uma situação de conflito.

Além disso, garantir a maior parte do tempo ARTs bem melhores às classes menos prioritárias, significa também um desperdício de recurso, pois o excesso de satisfação oferecido a estes usuários poderia ser melhor aproveitado para garantir maior satisfação aos usuários da classe *A* quando sua proporção crescer no sistema.

Quando o contrato de QoS é estabelecido de modo relativo, tal que é prometido a uma classe (mais exigente) um atendimento “melhor” em relação a uma outra menos exigente, a mesma situação denomina-se “inversão de prioridade” (Figura 1.1) e constitui uma falha no atendimento às especificações. Para o efeito da elaboração de modelos de negócios para provedores de serviços, seria conveniente investigar a possibilidade de evitar tal circunstância, atendendo, assim, às expectativas dos usuários.

1.3 Objetivos

O objetivo do presente projeto é o estudo e a análise de políticas de escalonamento RT, bem como a proposta de desenvolvimento de um novo método *Soft-RT*, a qual permita a obtenção de melhores resultados no atendimento das requisições pendentes nos provedores de serviço, evitando a ocorrência dos cenários de “*inversões de prioridade*” em termos de QoS relativa que ocorrem na arquitetura estudada por Monaco & Nobile (2009) e Casagrande et al. (2007).

Espera-se oferecer uma contribuição para o estado da arte na área de qualidade de serviço em nível de aplicação, para sistemas computacionais de tempo real, ampliando os resultados já obtidos nessa linha de pesquisa, visando à integração de QoS relativa e absoluta em uma mesma política de escalonamento.

1.4 Organização do Documento

Este trabalho de pesquisa apresenta a seguinte organização:

- No presente capítulo foram apresentadas considerações iniciais, a motivação para o desenvolvimento do trabalho, assim como os objetivos pertinentes para seu desenvolvimento e a organização do documento;

- No **capítulo 2** são introduzidos os conceitos de qualidade de serviço. Em seguida, aborda-se a QoS em nível de rede, descrevendo as arquiteturas de serviços integrados e de serviços diferenciados. Por fim, destaca-se a importância da QoS em nível de aplicação. Além disso, são abordados, de forma sucinta, algumas pesquisas realizadas sobre Qualidade de Serviço, evidenciando os esforços realizados por diversos pesquisadores nesse campo;
- No **capítulo 3** é apresentada a fundamentação teórica que embasa sistemas de tempo real, definindo suas principais classificações quanto ao sistema e quanto ao escalonamento, apresentando alguns algoritmos difundidos nessa área, incluindo a política EBS, o qual motivou a presente proposta;
- No **capítulo 4** é introduzido o conceito de simulação de sistemas, assim como a técnica para análise do modelo e do método propostos, descrevendo a teoria a partir da qual ela é fundamentada e familiarizando o leitor sobre a utilização de simulação de sistemas;
- No **capítulo 5** é apresentada a proposta e o desenvolvimento de um novo método para integração de QoS relativa e absoluta, sendo seu desempenho testado e avaliado por meio da realização dos experimentos e da obtenção dos resultados.
- No **capítulo 6** é apresentada a proposta de uma nova abordagem de modelo de sistemas para a política EBS composto por n classes de serviços, sendo seu desempenho testado e avaliado por meio da realização dos experimentos e comparação dos resultados obtidos com o modelo de duas classes de serviços apresentado pela política EBS.
- No **capítulo 7** é apresentada uma síntese das conclusões gerais, bem como as principais contribuições do presente trabalho e propostas para trabalhos futuros.

Qualidade de Serviço

2.1 Considerações Iniciais

No modelo atual de serviços da *Internet*, as requisições são tratadas, de forma geral, de modo equivalente, ou seja, a entrega dos dados é realizada de modo igual a todos os usuários. Nesse serviço, cada usuário compartilha a capacidade de transmissão com todos os fluxos de dados, os quais percorrem uma rota possível para chegar ao seu destino, de acordo com as rotas definidas e a capacidade de banda que estiverem disponíveis no momento.

Na ocorrência de congestionamento, não há garantia de desempenho bem como de sucesso na realização do serviço. Nesses casos, atrasos ou descartes de pacotes de dados podem ocorrer (Vasiliou, 2000).

Sendo assim, diante da expansão da *Internet* e da diversificação de suas aplicações e conseqüentemente de aumento no volume de dados, esse modelo não atende as necessidades dos usuários quanto a qualidade no serviço utilizado (Silva et al., 2006). Seria, então, conveniente, a diferenciação desses usuários. Portanto, a utilização do paradigma de **Qualidade de Serviço** torna-se cada vez mais importante e uma necessidade para a evolução da *Internet*, principalmente no segmento da utilização comercial (Bhatti et al., 2000; Strnadl, 2002).

Qualidade de Serviço pode ser definida de várias maneiras na literatura (Casagrande et al., 2007), de acordo com suas aplicações. De modo genérico, QoS pode ser entendida como a capacidade de fornecer a um elemento de rede algum nível de garantia de que seus requisitos de serviço serão satisfeitos (Zhao et al., 2000). Mais formalmente, a ISO

(*International Organization for Standardization*) define QoS como efeito coletivo do desempenho de um serviço, o qual determina o grau de satisfação de um usuário (ISO/IEC, 1998), ou seja, a percepção do usuário quanto à eficiência de um serviço (Peixoto, 2008).

No contexto das redes de computadores, embora o conceito de Qualidade de Serviço tenha aplicação em todos os níveis das arquiteturas convencionais, partes do desenvolvimento nesse campo têm se dado ao estudo e à elaboração de técnicas aplicáveis às camadas inferiores, especialmente na camada de rede (**QoS em Nível de Rede**).

Contudo, pode-se observar que os objetivos das técnicas de provisão de QoS implementadas nas camadas inferiores da arquitetura de redes seriam mais efetivas se refletissem também, na colaboração das camadas superiores, para que em situações de sobrecarga, as requisições não sejam tratadas de forma indiscriminada, o que não contribui para os esforços empreendidos ao longo da rede. Dessa forma, atualmente, tem havido um crescente interesse na provisão de **QoS em Nível de Aplicação** e é sob esse ponto de vista que se desenvolve este trabalho.

2.2 QoS em Nível de Rede

Em nível de rede, QoS pode ser definida como a capacidade de um elemento em prover certo grau de garantia do cumprimento dos requisitos de seu tráfego e serviços oferecidos, tais como níveis mínimos de perdas e atrasos de dados requisitados pelo usuário.

Dentre as várias abordagens existentes para provisão de QoS na *Internet*, destacam-se duas delas desenvolvidas sob a coordenação da IETF (*Internet Engineering Task Force*): a de **Serviços Integrados** (*IntServ - Integrated Services*) (Braden et al., 1994) e a de **Serviços Diferenciados** (*DiffServ - Differentiated Services*) (Blake et al., 1998).

2.2.1 Serviços Integrados

A abordagem de Serviços Integrados é caracterizada pela reserva de recursos, que consiste em alocar previamente os recursos de rede necessários para que se obtenha uma boa qualidade na transmissão de dados das aplicações. Para tanto, tais aplicações utilizam o protocolo RSVP (*Resource Reservation Protocol*), um protocolo de controle e sinalização atuante na camada de rede da pilha de protocolos TCP/IP. O funcionamento do RSVP é ilustrado pela Figura 2.1.

Conforme pode ser observado pela Figura 2.1, para a requisição dos recursos, um sistema emissor (*sender*) envia uma mensagem *PATH* ao receptor especificando os recursos que necessita. Cada roteador intermediário (*router*) repassa a mensagem para o próximo roteador, de forma que a mensagem chegue ao receptor (*receiver*). Após receber a mensagem *PATH*, o receptor envia uma mensagem *RESV* de volta ao emissor pelo caminho contrário da mensagem *PATH*, tentando, dessa forma, solicitar a reserva de recursos para

o fluxo de dados. Cada roteador intermediário pode aceitar ou rejeitar a reserva de recursos solicitada, de acordo com a quantidade de recursos disponíveis. Caso a requisição seja rejeitada, uma mensagem de erro é enviada de volta ao receptor e a sinalização é encerrada. Caso contrário, os recursos necessários para o fluxo são alocados e as informações de estado do fluxo de dados são armazenadas no roteador (Zhao et al., 2000).

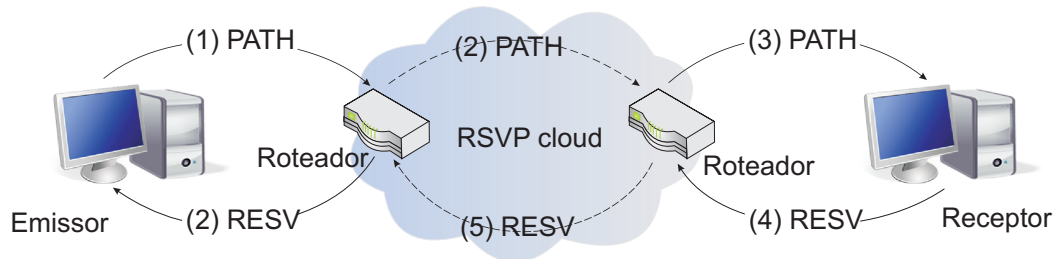


Figura 2.1: Ilustração do Funcionamento de Sinalização do *RSVP*. Adaptada de Zhao et al. (2000)

No entanto, com o aumento de fluxos, aumenta-se, consideravelmente, a quantidade de informações de estado, devido a reserva de recursos ser realizada para cada fluxo individualmente. Sendo assim, são necessários uma grande capacidade de processamento e um elevado poder de processamento dos roteadores, tornando o núcleo da rede mais complexo e, conseqüentemente prejudicando a escalabilidade da *Internet*. Além disso, todos os roteadores ao longo da rota devem oferecer suporte a serviços integrados. Tais problemas acabam dificultando a implementação e a implantação desse tipo de abordagem. De forma a solucionar os problemas mencionados, surge uma abordagem alternativa, denominada serviços diferenciados.

2.2.2 Serviços Diferenciados

A abordagem de Serviços Diferenciados é caracterizada pela definição de tipos de serviços (Magalhães & Cardozo, 1999). Sendo assim, o gerenciamento pelos roteadores torna-se mais simples, pois os fluxos são agregados e os roteadores não precisam guardar o estado de cada fluxo individualmente, como é o caso da abordagem de Serviços Integrados.

No cabeçalho de um pacote IP (*Internet Protocol*), os tipos de serviços podem ser representados por um campo denominado TOS (*Type of Service*). Define-se um novo *layout* (DSField - *Differentiated Service Field*) para esse campo, conforme descrito em Nichols et al. (1999). Com base nesse *layout*, um conjunto de procedimentos distintos de envio de pacotes é especificado, oferecendo, dessa forma, diferentes classes de serviços.

Em uma solicitação de serviço diferenciado, é importante o estabelecimento de um contrato de nível de serviço, firmado entre o usuário e o provedor de serviços de *Internet* (ISP - *Internet Service Provider*). Esse contrato, denominado SLA (*Service Level Agreement*) determina as classes de serviços suportadas e a quantidade de tráfego na banda

entre os domínios.

Contratos SLA podem ser definidos como estáticos ou dinâmicos. Um SLA é considerado estático quando negociado de forma regular (mensalmente ou anualmente). Já em um SLA dinâmico, os serviços podem ser solicitados sob demanda. Sendo assim, faz-se necessária a utilização de algum protocolo de sinalização e controle (como o RSVP) de forma a indicar as necessidades do usuário em determinados momentos (Zhao et al., 2000).

Portanto, a abordagem empregada pelo *DiffServ* baseia-se em um esquema de prioridades relativas, ou seja, ele não garante que o tráfego gerado por uma aplicação alcançará o seu destino dentro de um limite máximo de tempo, ou com uma variação máxima de atraso, mas sim que um tráfego com um certo nível de prioridade receberá um atendimento melhor que qualquer outro que possua uma prioridade inferior (Xiao & Ni, 1999).

Tal abordagem foi utilizada em Chen & Heidemann (2003), no qual é proposto um algoritmo denominado SFD (*Short Flow Differentiating*) com o intuito de melhorar o desempenho de tráfego *Web* interativo. O algoritmo proposto prioriza tráfegos mais curtos, como os de rajada, visto que, estudos mostraram que tráfegos mais curtos correspondem a mais de 80% das respostas *Web*, além de fluírem mais rapidamente pela rede. Os resultados apresentados, por meio de simulação, mostraram a importância e o impacto na priorização de serviços mais curtos, com uma melhora de mais de 30% no tempo de resposta das transmissões mais curtas.

2.3 QoS em Nível de Aplicação

Na seção anterior, foram abordadas técnicas de garantia de QoS em nível de rede, particularmente as abordagens de serviços integrados e diferenciados.

Entretanto, a provisão de QoS seria mais abrangente e propiciaria melhores resultados se houvesse a cooperação de todos os elementos do sistema, o que implica na consideração de garantias de **QoS em nível de aplicação**, a qual define a forma de atendimento do sistema a determinada requisição.

Quando a forma de atendimento do sistema é definida com uma relação de prioridade como base do contrato de serviço, tem-se a **QoS relativa**. Nessa abordagem, com a diferenciação entre classes garante-se um melhor atendimento a uma determinada classe mais prioritária, cujo valor estabelecido em contrato é menor em relação às demais classes. No caso específico de tempo de resposta, um contrato de QoS relativa especifica que as requisições provenientes de determinada classe serão atendidas em geral antes daquelas provenientes de uma classe menos prioritária.

Quando a forma de atendimento, porém, é definida com o estabelecimento de metas e valores de desempenho a serem respeitadas para cada classe individualmente, tem-se a **QoS absoluta**. Nessa abordagem, existe a garantia por classe, independente do que é definido para as demais. Nesse caso, a prioridade é definida dinamicamente em função do

estado do sistema e das condições efetivas dos contratos. São estabelecidas taxas mínimas de serviço ou atrasos máximos de atendimento para as requisições. No caso específico de tempo de resposta, um contrato de QoS absoluta especifica limites para o tempo de resposta das requisições de cada classe, independente daquele praticado para as outras.

Alguns trabalhos têm sido desenvolvidos com o intuito de viabilizar e evoluir as formas de disponibilização de QoS em nível de aplicação, tanto em termos relativos quanto em termos absolutos (Almeida et al., 1998; Eggert & Heidemann, 1999). Diversos pesquisadores têm realizado esforços particularmente no contexto de servidores *Web* (Abdelzaher et al., 2002; Casagrande, 2007; Kanodia & Knightly, 2003; Lee et al., 2003; Monaco et al., 2009; Pan et al., 2008; Pradhan et al., 2002; Wei et al., 2005). Nesse contexto as pesquisas mais preponderantes abordam: controle de admissão e diferenciação de serviço (Estrella et al., 2006; Messias et al., 2007; Teixeira et al., 2005; Traldi et al., 2006). As próximas seções permitem a análise de alguns desses trabalhos.

2.3.1 QoS Relativa

Trabalhos abordando QoS relativa, mais freqüentes na literatura, capturam a noção intuitiva de atendimento preferencial comum aos modelos de prestação de serviço convencional de provedores comerciais, oferecendo classes de serviço com prioridades.

Kanodia & Knightly (2003), por exemplo, propõem um modelo de serviço *Web* baseado na diferenciação de classes e no controle de admissão de forma a gerenciar os tempos médios de resposta contratados e maximizar a utilização de recursos. Sendo assim, além de tentar manter valores individuais das classes, a técnica proposta também realiza um controle adaptativo de tempos de resposta entre as classes. O modelo proposto é validado por meio de simulação orientada a *traces*¹. Os resultados mostraram que o algoritmo, por realizar abstração de recursos em alto nível, pode ser utilizado em conjunto com sistemas operacionais dotados de mecanismos de QoS, e explorando suas funcionalidades oferecer altas taxas de utilização de servidor, bem como satisfazer as especificações de serviços das várias classes.

Inspirado no modelo *DiffServ* pertinente ao nível de rede, Teixeira et al. (2005) realiza uma transposição de seus princípios para a camada de aplicação, concebendo um modelo de servidor *Web* com diferenciação de serviços (SWDS). Esse tipo de abordagem está relacionada à QoS relativa, usualmente mais comum em servidores *Web* (Henriksson et al., 2004; Kang et al., 2003; Lu et al., 2001). A arquitetura de servidor *Web* proposta fornece serviços diferenciados a seus clientes de acordo com suas características de demanda. Nessa arquitetura são propostos mecanismos de diferenciação de serviços baseados em prioridades e um módulo de controle de admissão com o objetivo de garantir estabilidade

¹Registros de eventos de um sistema ordenado de acordo com o tempo. Simulação orientada a *traces* utilizam tais registros para gerar sua entrada (como por exemplo, *traces* de escalonamento de *jobs* e *logs* de servidores *Web*).

na qualidade de serviço oferecida. A arquitetura basicamente é composta por um módulo classificador de requisições, por um controle de admissão e um *cluster* de servidores *Web*. Quando admitida, a requisição é atribuída a um dos nós do *cluster* segundo o algoritmo de escalonamento ou diferenciação de serviços vigente.

Outra contribuição desse trabalho (Teixeira et al., 2005) é a proposta e análise de um mecanismo de diferenciação de serviços denominado PRIAdap (Prioridades Adaptativo), o qual atribui uma maior ou menor importância às requisições de alta prioridade, a fim de evitar que estas venham a monopolizar o uso de uma dada classe percorrendo um certo número k de posições na fila. Esse parâmetro é denominado *look-ahead*, o qual permite regular o nível de priorização do sistema, determinando assim o quão rigoroso será o esquema de prioridades empregado. A arquitetura e os mecanismos são validados por meio de simulação.

Em Estrella et al. (2006) estende-se o trabalho realizado em Teixeira et al. (2005), incluindo um mecanismo de negociação ao módulo de controle de admissão do modelo SWDS, com o objetivo de permitir que requisições de determinadas classes de serviços tenham prioridade no atendimento em relação às classes inferiores, mesmo em situações de sobrecarga nos servidores *Web*. Para tanto, são propostos dois algoritmos: Algoritmo de Negociação Rigoroso (ANR) e Algoritmo de Negociação com o Cliente (ANC).

O algoritmo ANR possibilita uma nova oportunidade para que requisições de determinadas classes de serviços possam ser readmitidas no sistema e, futuramente, possam ser atendidas; porém os requisitos de QoS são rebaixados. Já no algoritmo ANC, a requisição não admitida retorna ao cliente e esse por sua vez, decide se aceita ou não o rebaixamento da QoS. Os resultados, obtidos por simulação, permitem afirmar que mecanismos de negociação são importantes quando se trabalha em projeto de arquitetura de servidores *Web* com diferenciação de serviço. Tais mecanismos proporcionaram melhores médias de tempo de resposta e menores taxas de descarte de requisições, melhorando, dessa forma, a qualidade de serviço oferecida. Dependendo do algoritmo (em especial, o de negociação com a participação do cliente) a porcentagem de descarte para classes prioritárias torna-se muito menor.

Em Traldi et al. (2006) também é abordada a QoS em termos relativos, propondo e comparando dois novos algoritmos de diferenciação de serviços com o intuito de prover qualidade de serviço em servidores *Web*, sendo eles o Reserva Adaptativa de Recursos (RSVAdap) e o *Weighted Fair Queuing* (WFQ). Esses algoritmos foram adicionados ao modelo proposto em Teixeira et al. (2005) e simulados sem a utilização de políticas para o controle de admissão e em sistemas homogêneos e heterogêneos.

O algoritmo RSVAdap é baseado na Reserva de Recursos, realizando a alocação de recursos de forma dinâmica, ou seja, sob demanda, segundo a carga de trabalho vigente. O controle da alocação se baseia no número de requisições de cada classe presente no sistema e no nível de diferenciação pretendido. Pretende-se com isso uma melhor utilização dos

recursos do sistema. O algoritmo WFQ é uma adaptação, para o nível de aplicação, do *Weighted Fair Queuing* existente no nível de rede. Além de oferecer diferenciação entre classes, esse algoritmo de escalonamento tem como característica a ausência da negação de serviço, como pode ocorrer nos mecanismos de Prioridade Rigoroso e Adaptativo (Teixeira et al., 2005). Ele consiste na divisão da capacidade de processamento dos nós entre as classes de requisições, de acordo com os pesos que são atribuídos dinamicamente às classes.

Em outro trabalho (Messias, 2007) é apresentado um estudo, implementação e avaliação do modelo de servidor *Web* com diferenciação de serviços. Com o intuito de controlar a carga no sistema, algoritmos de reserva de recursos, escalonamento baseado em prioridades e mecanismos de controle de admissão, foram considerados. Embora os algoritmos de reserva de recursos (*RSV* e *RSVAdap*) sejam eficientes para provimento de diferenciação entre as classes consideradas, seus desempenhos não foram satisfatórios em algumas situações, devido a arquitetura em que foram implementados e por motivos inerentes ao próprio algoritmo. O algoritmo de escalonamento baseado em prioridades (*PriProcess*), mostrou-se mais eficiente tanto na obtenção de diferenciação de serviço entre as classes, como na obtenção de desempenho. Foi desenvolvido também um mecanismo de controle de admissão com diferenciação de serviços. Os resultados obtidos indicam uma melhora em termos de tempos de respostas e número de requisições completadas para a classe de maior prioridade.

2.3.2 QoS Absoluta

Mais recentemente têm sido desenvolvidas pesquisas sobre QoS em termos absolutos, com a qual se pode oferecer garantias mais estritas de qualidade, conforme pode ser observado em Casagrande (2007); Monaco et al. (2009); Nery (2008); Peixoto (2008); Tott (2008); Wei et al. (2005).

Em Wei et al. (2005) é proposta uma abordagem de controle *fuzzy* para garantir atrasos absolutos em servidores *Web*. A arquitetura do sistema proposto é composta pelos módulos: escalonador de conexão, monitor de sistema e controlador *fuzzy*.

O escalonador de conexão é responsável pelo recebimento e aceitação de todas as requisições de conexão que chegam ao sistema. Essas requisições são classificadas em diferentes classes baseadas, por exemplo, em endereços IP dos usuários. Uma nova requisição é alocada a um processo somente se o número de processos atribuídos à determinada classe de requisições é menor que o contador de processos dessa classe.

O monitor é responsável pela medição e informação ao sistema do atraso absoluto de cada classe de requisições.

O controlador *fuzzy*, por sua vez, é responsável pelo ajuste do contador de cada classe (cálculo do número de processos que são alocadas para cada uma delas), de forma que o atraso das classes não ultrapasse o limite estipulado.

Um estudo realizado em Casagrande (2007) introduz uma política de escalonamento de tempo real não-determinístico (*Soft-RT*) para provisão de garantias de tempo de resposta estocásticas em ambientes interativos *online*, mais especificamente em servidores *Web*.

No modelo proposto, são definidas duas classes de usuários (A e B), as quais apresentam menores e maiores valores (tempos médios de resposta) estabelecidos em contrato, respectivamente. O parâmetro de QoS contratado por cada usuário é um limite superior para a média do tempo de resposta que é especificado previamente por um acordo entre o provedor de serviços e o usuário.

A política elaborada tem como princípio comparar tal limite com o tempo de resposta médio efetivo, que reflete a média instantânea de tempo de resposta de fato praticada, o que possibilita a verificação dos contratos que estejam mais próximos de serem violados.

Casagrande (2007) utiliza uma estratégia híbrida associada à heurísticas baseadas nos algoritmos clássicos EDF (*Earliest Deadline First*) e SJF (*Shortest Job First*) (ver Seção 3.3.2), levando em consideração, portanto, o tempo de espera em fila (urgência de *deadline*) e o custo de execução de uma requisição, impondo ao sistema a menor demanda de recursos possível e garantindo um compromisso entre desempenho e confiabilidade no atendimento de contratos individuais.

Nesse trabalho foi demonstrado que a política EBS proporciona resultados superiores à heurísticas convencionais. Esses resultados, bem como o funcionamento detalhado dessa política podem ser observados na Seção 3.3.2.

Em Peixoto (2008) é desenvolvido um estudo no qual estende-se a abordagem de Casagrande (2007), apresentando e comparando políticas de escalonamento que têm por objetivo prover QoS absoluta para um *array* de servidores *Web* heterogêneos. Para tanto, considera-se uma arquitetura de **escalonamento ortogonal**, na qual além da política de **escalonamento de requisições** atuar na ordenação da fila de *jobs*, uma política de **escalonamento de recursos** atua atribuindo-se o *job* ao recurso (processador) em que será executado.

Em seu trabalho, Peixoto (2008) compara um conjunto de combinações de políticas de fila (EDF e EBS) e políticas de recursos (*Multiple Queue* - MQ, *Single Queue* - SQ e *Dynamic Single Queue* - DSQ).

No modelo MQ ou *Web switch*, as requisições são enviadas para um servidor qualquer do *array* por meio do balanceamento de carga, de acordo com algumas regras de escalonamento, tais como políticas que levam em consideração as características de carga do servidor.

Os modelos SQ e DSQ apresentam apenas uma única fila sendo gerenciada de maneira centralizada. No modelo SQ, a fila de requisições é ordenada baseando-se nos resultados obtidos com uma heurística, sendo o primeiro *job* executado no primeiro servidor livre do *array*.

No modelo DSQ, implementa-se um modelo híbrido entre SQ (com fila única) e o

Web switch (com balanceamento). Nesse modelo, ao invés de atribuir a requisição mais prioritária ao primeiro servidor livre, o DSQ segue o mesmo princípio da EBS, prevendo o impacto causado ao sistema. Sendo assim, para cada requisição da fila de requisições, o DSQ seleciona o servidor que apresente o menor tempo estimado de término de processamento da requisição em questão. Caso o servidor esteja ocupado, a requisição é colocada em uma das filas virtuais, referente ao seu respectivo recurso.

A avaliação de desempenho da arquitetura ortogonal demonstra que a mesma obtém um bom desempenho na provisão de QoS absoluta com relação a mudanças instantâneas das cargas de trabalho no ambiente *Web*. Os resultados experimentais demonstram que a combinação DSQ-EBS, combinação da EBS (na política de fila) com a política de recurso, proposta nesse trabalho, é superior às outras combinações examinadas.

O modelo DSQ é o que melhor se adaptou às características da política de fila EBS, principalmente em cenários que apresentam uma alta variação contratual, devido a propriedade de balanceamento inerente ao escalonador.

Em outro trabalho, associado aos anteriores, Tott (2008) investiga o impacto da carga gerada por cada usuário no sistema como um todo e como essa carga influencia o serviço oferecido a outros usuários. Sendo assim, apresenta-se um mecanismo de controle de admissão de requisições, capaz de administrar o nível de degradação do sistema, isolando o efeito do comportamento de um usuário sobre a qualidade de serviço oferecida aos demais.

Para tanto, foram propostas algumas melhorias à política EBS, uma delas é a existência de **contratos bilaterais**, os quais limitam o comprometimento do sistema em atender os usuários na média determinada de tempo. De acordo com esses contratos, a garantia de atendimento do sistema ao usuário ocorrerá como previsto, caso a carga imposta pelo mesmo ao sistema não ultrapassar um limiar pré-estabelecido.

Outra melhoria, refere-se à redução de ordem de complexidade da política EBS. Tott (2008) propõe um algoritmo de escalonamento de requisições utilizando uma estrutura do tipo *Heap*, com o intuito de diminuir a complexidade do algoritmo linear $O(n)$ para logarítmica $O(\log(n))$, sem impactar negativamente no atendimento das requisições. O algoritmo proposto mostra-se útil em cenários onde o número de requisições ao sistema seja extremamente elevado, e portanto o tamanho da fila de requisições seja consideravelmente grande. Sendo assim, com a redução de ordem de complexidade do algoritmo, o tamanho da fila de requisições deixa de ser um empecilho ao crescimento e escalabilidade do sistema.

Embora a meta de manter o tempo médio de resposta efetivamente abaixo do limite estipulado em contrato tenha sido alcançada nos experimentos realizados em Casagrande (2007), considerações acerca da aplicabilidade do método em situações práticas, suscitam conjecturas sobre as condições em que contratos de QoS dessa natureza sejam suficientes para as necessidades ou expectativas dos usuários da aplicação *Web*, visto que tais contratos não apresentam nenhuma restrição referente à dispersão do tempo médio de resposta

efetivo.

Considerando-se duas situações de dispersão: a primeira delas, em que os tempos de resposta efetivos, em uma determinada **janela** considerada, estejam dispersos em torno da média, formando uma distribuição normal; a segunda, em que os tempos de resposta estejam concentrados em duas regiões distantes da média, uma à direita e outra à esquerda desta.

Em ambas as situações, o tempo médio de resposta efetivamente praticado pode ser o mesmo, indicando que o contrato foi atendido. No entanto, na primeira situação o usuário recebe uma qualidade de serviço aproximadamente constante ao longo do tempo, enquanto que na segunda, há momentos em que a qualidade oferecida pode exceder a expectativa dos usuários, e momentos onde o serviço é percebido como muito ruim. Isso acontece, por exemplo, quando um usuário é atendido com uma média muito inferior à contratada, durante um determinado período em que a carga do sistema esteja baixa, o que resulta em uma grande folga em seu contrato, permitindo à regra de tomada de decisão, eventualmente praticar reiteradamente altos tempos de resposta para o usuário, sem comprometer sua média efetiva.

Sendo assim, Nery (2008) investiga não apenas o controle do tempo médio de resposta efetivamente praticado aos usuários, mas também a **dispersão** dos tempos de atendimento em torno dele.

2.4 Considerações Finais

Devido à crescente utilização e diversificação das aplicações suportadas pela *Internet*, embora seu modelo atual de serviços tenha funcionado e ainda funcione bem para vários tipos de aplicações, seria conveniente a utilização do modelo de **Qualidade de Serviço**. Como visto (seções 2.2.1 e 2.2.2), as abordagens para provisão de QoS na *Internet* mais utilizadas são as de **Serviços Integrados** e de **Serviços Diferenciados**. Além da **QoS em nível de rede**, foi evidenciada a importância do fornecimento de **QoS em nível de aplicação**, assunto esse, tema principal do projeto de pesquisa em questão, bem como alguns trabalhos mais relevantes da área estudada, tanto para embasamento teórico para o desenvolvimento de alternativas que venham a complementar as soluções já propostas, como também para orientar sobre a necessidade de pesquisas em campos pouco explorados.

Esforços em outras áreas, como Tempo Real, também têm sido realizados com o intuito de propor novas abordagens para o problema, sendo importante uma breve revisão de seus principais conceitos (ver capítulo 3).

Sistemas de Tempo Real

3.1 Considerações Iniciais

Com a crescente utilização de sistemas computacionais interativos na realização de serviços, requisitos de desempenho e confiabilidade associados à responsividade de aplicações como ensino a distância, telemedicina, comércio eletrônico, dentre outros, tornam-se essenciais à efetividade desses serviços (Monaco et al., 2009; Xiangbin, 2008).

O estudo e o desenvolvimento de técnicas para a análise e síntese de mecanismos que garantam atendimento de restrições temporais correspondem ao domínio de **sistemas de tempo real**.

Em ambientes computacionais onde as noções de tempo são tratadas sob a forma de qualidade de serviço e são especificadas explicitamente, conceitos e técnicas de escalonamento desempenham um papel preponderante no comportamento de sistemas de tempo real (Sha et al., 2004). Dessa forma, tais conceitos e técnicas serão explicitados nas seções subsequentes 3.2 e 3.3.

3.2 Sistemas de Tempo Real

Abordagens de tempo real são utilizadas em sistemas que necessitam sincronizar eventos internos com o ambiente (eventos externos ao sistema). Para tanto, é necessário satisfazer as requisições temporais individuais, garantindo que cada requisição seja atendida antes do seu *deadline* (restrição de tempo, correspondendo ao instante máximo em que uma

requisição deve ser concluída) (Cheng, 2002), de forma a não violar a sincronização do sistema com o ambiente externo (Nissanke, 1997).

Tais sistemas não necessariamente implicam em desempenho computacional, mas sim em garantir o tempo de resposta do sistema aos eventos externos (Nissanke, 1997).

Portanto, o requisito-chave na formulação das especificações RT é o limite superior para atrasos no tempo de reação do sistema aos estímulos externos, ou seja, restrições sobre máximos tempos de resposta.

Para a descrição e caracterização dos diferentes tipos de sistemas RT, bem como os métodos para escalonamento e gerência de recursos, utilizam-se termos gerais para tratamento da carga de trabalho de sistemas computacionais e de comunicação (Casa-grande, 2007). Sendo assim, a unidade de trabalho agendada e executada por um sistema, denomina-se requisição (ou *job*), já o conjunto dessas unidades, o qual realiza determinadas funções no sistema será denominado, neste trabalho, tarefa (ou *task*) (Liu, 2000).

3.2.1 Sistemas *Hard-RT*, *Soft-RT* e *Firm-RT*

Os sistemas de tempo real podem ser classificados em três tipos: *Hard Real-Time*, *Soft Real-Time* e *Firm Real-Time* (Mathes et al., 2008), os quais diferenciam-se segundo a forma de restrição de tempo estabelecida pelo sistema a uma determinada tarefa (Laplante, 2004).

Para **sistemas *Soft Real-Time***, a precisão de tempo é importante, porém não é crítica. Um atraso na conclusão de uma tarefa é indesejável, mas é aceitável, pois não trará sérios danos, apenas uma degradação do desempenho a cada não cumprimento de *deadline*. Nesses sistemas, as tarefas são realizadas tão rapidamente quanto possível, podendo haver um certo relaxamento na precisão de alguns de seus tempos em situações de sobrecarga do sistema, conforme pode ser observado em Calandrino et al. (2007).

Calandrino et al. (2007) propõem uma abordagem para escalonamento de tarefas periódicas (ver Seção 3.2.3) *Soft-RT* em uma arquitetura AMP¹ (*Asymmetric Multicore Platforms*) (Calandrino et al., 2007), com o intuito de prover um bom desempenho às tarefas não RT, na presença de carga de trabalho RT. Para tanto, requisições dessa carga, uma vez completado seus processamentos respeitando seus respectivos *deadlines*, não recebem benefícios para melhorar seus tempos de resposta. Sendo assim, assume-se que uma tarefa não RT pode apresentar uma maior prioridade em relação às outras tarefas, quando tal fato não causar violações de *deadline* das requisições RT. Vale lembrar que isso é possível por se tratar de requisições *Soft-RT*.

Em **sistemas *Hard Real-Time***, é de extrema importância a exatidão do tempo de resposta. Nesses sistemas, o não cumprimento das restrições temporais (como tempo de

¹Tal arquitetura consiste em um grande número de unidades de processamento em um ou mais *chips*, sendo que, cada unidade de processamento é capaz de executar a mesma instrução a um nível de desempenho diferente.

liberação e *deadline*) pode ocasionar graves conseqüências (Zhang et al., 2008), como o comprometimento da confiabilidade do sistema ou até mesmo colocar vida de pessoas em risco (Liu, 2000; Tavares et al., 2008). Por serem mais desafiadores, sistemas *Hard-RT* apresentam um custo mais elevado em relação a sistemas *Soft-RT* (Mathes et al., 2008).

Já em sistemas *Firm-RT*, como em sistemas *Soft-RT*, o não cumprimento das restrições temporais não necessariamente implica em falha. Esses sistemas toleram determinadas frações de atrasos, porém, diferentemente de sistemas *Soft-RT*, o resultado gerado depois de ultrapassado o *deadline* torna-se inútil, sendo assim, as requisições que encontram-se fora do prazo são descartadas. Um exemplo de sistemas *Firm-RT* seria a transmissão de áudio e vídeo.

3.2.2 Restrições Temporais

Restrições temporais são relevantes, tanto para caracterizar o sistema, quanto para impor o comportamento temporal desejado ou necessário de uma requisição do sistema, durante a execução das tarefas, conforme pode ser observado em Mathes et al. (2008) e Zhang et al. (2008). Dentre elas (Figura 3.1), além do *deadline* (d), têm-se (Leite, 2007):

- **Tempo de chegada** (*arrival time* - at): correspondente ao tempo em que o escalonador tem conhecimento da ativação de uma requisição.
- **Tempo de liberação**: (*release time* - rt) correspondente ao instante em que a requisição está disponível para processamento e portanto é incluída na fila de prontas para serem executadas;
- **Release jitter** (J): correspondente à variação do retardo na entrega de dados, ou seja, a medida de variação do atraso entre as sucessivas requisições.
- **Tempo de início** (*start time* - st): correspondente ao instante inicial do processamento da requisição;
- **Tempo de execução** (*computation time* - c): correspondente ao tempo gasto para completar a execução de uma determinada requisição. Útil para determinar se uma requisição cumprirá seu *deadline*;
- **Tempo de término** (*completion time* - ct): correspondente ao instante final do processamento da requisição;
- **Tempo de resposta** (*response time* - r): correspondente ao tempo desde o recebimento de um estímulo externo pelo sistema (um conjunto de entradas) até a disponibilização de todas as saídas associadas (resposta);

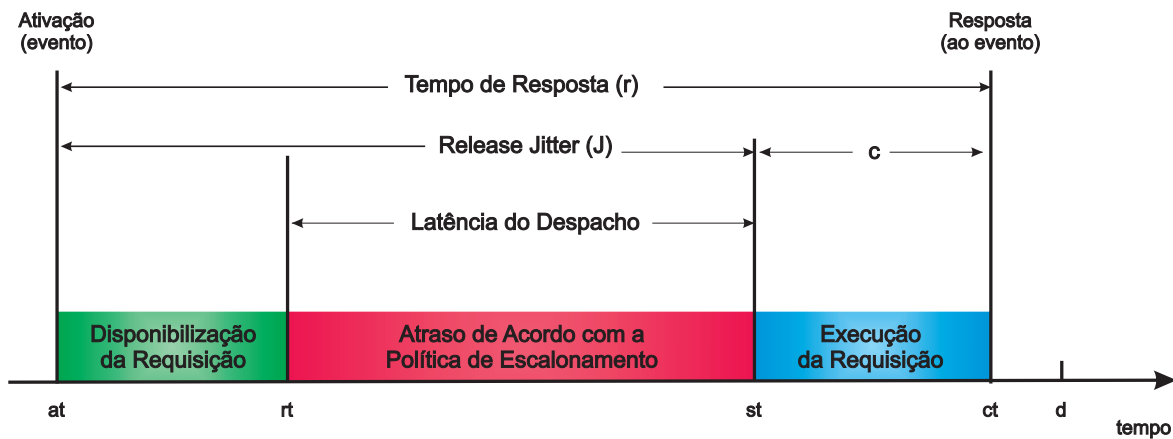


Figura 3.1: Ilustração das Restrições Temporais. Adaptada de Leite (2007).

3.2.3 Caracterização das Tarefas

Existem três tipos de tarefas em sistemas RT convencionais: **periódicas**, **esporádicas** e **aperiódicas**.

Uma tarefa (T_1) é considerada **periódica** (Figura 3.2 (a)) quando existem várias instâncias (ou iterações) da tarefa e as ativações (liberações consecutivas) ocorrem em um período fixo p (intervalo de tempo regular) (Lin & Tarng, 1991). Uma tarefa (T_2) é **esporádica** (Figura 3.2 (b)), quando se conhece apenas o intervalo mínimo (min) de tempo entre ativações de instâncias consecutivas. Já uma tarefa **aperiódica** (T_3) (Figura 3.2 (c)), nada mais é que uma tarefa esporádica com uma frequência de ativação de suas instâncias e duração indeterminadas (Cheng, 2002; Cruz & Lima, 2006).

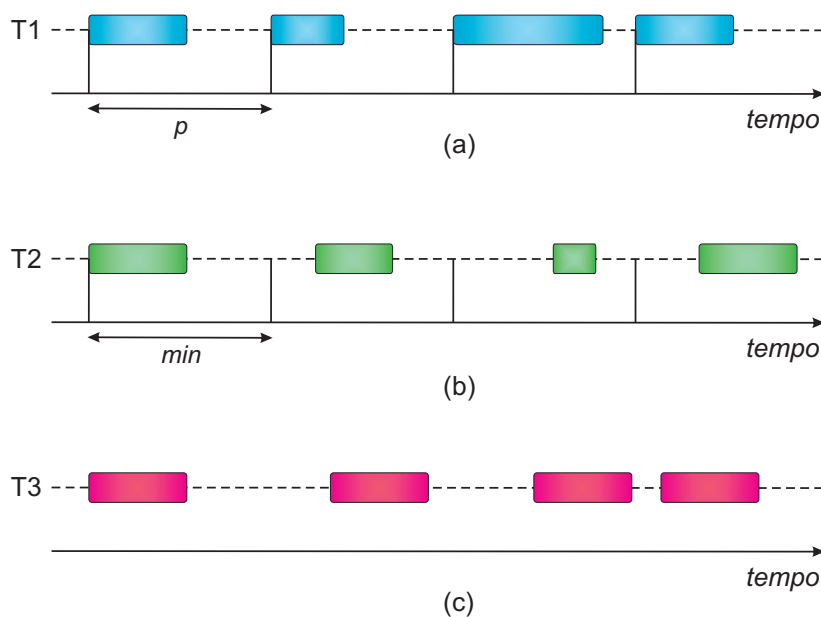


Figura 3.2: Caracterização das tarefas: (a) periódica; (b) esporádica e (c) aperiódica. Adaptada de Leite (2007).

3.3 Escalonamento em Sistemas RT

Em sistemas de tempo virtual², o escalonamento baseia-se em maximizar a soma das tarefas concluídas, minimizar a duração do escalonamento, minimizar o número de processadores requeridos ou minimizar o máximo de tempo de processamento. Entretanto, em sistemas de tempo real, a relevância do **escalonamento** é ainda mais evidente, visto que é importante a especificação da ordem temporal de alocação de requisições para um processador (ou um conjunto de processadores), de forma a respeitar as restrições de tempo de cada uma delas e a disponibilidade de recursos (Cheng, 2002).

3.3.1 Classificação do Escalonamento

O escalonamento pode ser classificado de acordo com suas características, como **preempção**, **natureza do problema** e **tipo de abordagem**.

Sistemas de tempo real são, muitas vezes, projetados, utilizando-se **escalonamento com preempção**, de forma que uma requisição possa ser interrompida para garantir a execução de outras mais prioritárias e, continuada sem qualquer comprometimento em sua realização.

Embora ofereça mais possibilidades de escalonamento, a utilização de preempção pode demandar um tempo maior de execução devido a necessidade de troca de contexto. Além disso, o custo da preempção pode ser bastante elevado, devido a alguns problemas de escalonamento de tempo real, tais como escalonamento de entrada e saída.

Dessa forma, Andersson & Tovar (2006) propõe um algoritmo com o intuito de minimizar o número de preempções. Outros trabalhos, como os apresentados em Kermia & Sorel (2008) e Jeffay et al. (1991), têm sido realizados, utilizando-se **escalonamento sem preempção**, de forma que, uma vez iniciada a execução de uma requisição, não ocorrem interrupções até o seu término.

Considerando a **natureza do problema**, o escalonamento pode ser classificado em: **estático**, **dinâmico** e **misto**. O escalonamento **estático** apresenta um conhecimento prévio das características relevantes de todas as requisições. Já o escalonamento **dinâmico** trabalha com requisições imprevisíveis com relação ao tempo de chegada e, possivelmente com tempo de execução desconhecido. Um escalonamento **misto** é a combinação dos dois anteriores, escalonando estaticamente as requisições por meio de características previamente conhecidas e ajustando as outras à medida que chegam (Nissanke, 1997).

Com relação ao **tipo de abordagem**, o escalonamento RT se divide em: **orientada ao tempo**, **Weighted Round Robin** e **orientada à prioridade**. Na primeira abordagem, **orientada ao tempo** (*Clock-driven* ou *Time-driven*), decisões sobre qual requisição executar são tomadas, de acordo com instantes previamente estabelecidos para cada uma

²Um sistema que não é RT pode ser designado de tempo virtual.

antes da execução no sistema.

A abordagem **Weighted Round-Robin** é utilizada no escalonamento de aplicações de tempo compartilhado (*time-shared*), em que as requisições são organizadas de acordo com o modelo de fila FIFO (*First In First Out*) quando se tornam prontas para execução (ver Seção 3.3.2).

Na terceira abordagem, **orientada à prioridade**, o escalonamento baseia-se em eventos (*event-driven*). Considerando-se que uma requisição T_1 apresente uma maior prioridade em relação a outra T_2 , a requisição de menor prioridade (T_2) pode sofrer interrupções para a execução de uma nova requisição T_1 . Caso seja requerida, simultaneamente, mais de uma requisição com alta prioridade, aquela que apresentar maior prioridade dentre todas é executada. Se houver equivalência de prioridades, uma é escolhida arbitrariamente.

O escalonamento a ser focado neste trabalho será **sem preempção, dinâmico e orientado à prioridade**, dado sua importância na literatura e pelo fato de abranger os mais variados aspectos dos comportamentos temporais de aplicações RT.

Pode-se constatar que a forma com que o escalonamento ocorre é primordial à obtenção de produtividade e eficiência. Caso as tarefas não consigam atender seus *deadlines*, tal escalonamento não é uma solução viável (Stankovic et al., 1995). Portanto, deve-se utilizar uma política de escalonamento que permita a obtenção dos melhores resultados possíveis.

3.3.2 Algoritmos de Escalonamento

Em cenários específicos, diversos algoritmos de escalonamento têm sido desenvolvidos, como por exemplo em Kato & Yamasaki (2008) e Shi-jun et al. (2008). Alguns são muito importantes em função de sua larga aplicação e relevância científica, conforme pode ser observado em Leontyev & Anderson (2007), Devi & Anderson (2006b) e Devi & Anderson (2006a), dentre eles, pode-se destacar os algoritmos FIFO, SJF, EDF, bem como a política EBS (Casagrande et al., 2007), a qual motivou a presente proposta e possibilitou o prosseguimento de numerosas variações e adaptações (Monaco et al., 2009; Nery, 2008; Peixoto et al., 2008; Tott, 2008).

Algoritmo FIFO - *First In First Out*

A política de escalonamento mais utilizada do modelo convencional de servidor *Web* é a FIFO. Nessa política, as requisições são atendidas na ordem em que chegam ao sistema e que apresentam-se na fila de espera (Ye et al., 2005, 2007).

Para ilustrar essa situação, considera-se quatro requisições (R_1 , R_2 , R_3 e R_4) em uma fila e seus respectivos tempos de execução, representados em unidades de tempo (*u.t.*), *9u.t.*, *7u.t.*, *5u.t.* e *3u.t.*

A Figura 3.3 ilustra a ordem em que as requisições são processadas e os tempos de espera em fila de cada uma dessas requisições, utilizando-se a política FIFO. Sendo assim,

o tempo médio de resposta proporcionado por este escalonamento seria $(0+9+16+21)/4 = 11.5ut$.

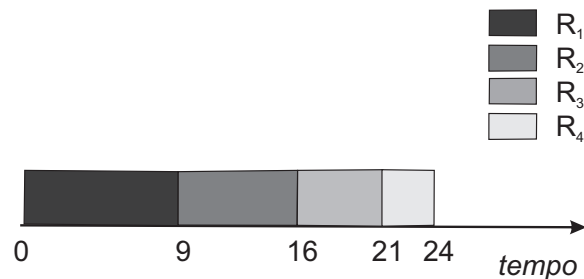


Figura 3.3: Exemplo da Utilização do Algoritmo de Escalonamento FIFO. Adaptada de Gomes (2009).

Algoritmo SJF - *Shortest Job First*

A utilização de outras políticas de escalonamento pode proporcionar menores tempos médios de resposta, dentre essas encontra-se a política SJF. Nessa política, as requisições são organizadas em uma fila de acordo com o tempo de processamento (T_p), sendo os menores *jobs* colocados à frente. Realocar as requisições mais curtas na frente das que exigem maior tempo de processamento permite que o tempo de espera das mais curtas diminua mais do que aumenta o das requisições mais longas, possibilitando uma diminuição na média de tempo de resposta oferecida pelo sistema.

A Figura 3.4 ilustra a ordem em que as requisições são processadas e seus respectivos tempos de espera, utilizando a política SJF. Dessa forma, o tempo médio de resposta é de $(0 + 3 + 8 + 15)/4 = 6.5ut$.

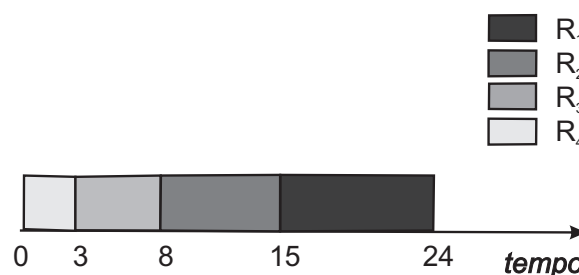


Figura 3.4: Exemplo da Utilização do Algoritmo de Escalonamento SJF. Adaptada de Gomes (2009).

Algoritmo EDF - *Earliest Deadline First*

No escalonamento de sistemas RT, é importante que os *deadlines* de todas as requisições sejam satisfeitas (Cheng, 2002). Um dos algoritmos mais utilizados, nesses sistemas, é o

EDF (Shi-jun et al., 2008). O escalonamento das requisições utilizando o algoritmo EDF, é realizada de forma dinâmica (em tempo de execução), atribuindo prioridades à elas de acordo com seus *deadlines*. A priorização das requisições baseia-se, portanto, na urgência das requisições. Sendo assim, a requisição mais prioritária é a que apresenta o *deadline* mais próximo do tempo atual. A cada nova requisição que chega, a fila de requisições prontas é reordenada, atualizando a nova distribuição de prioridades (Farines et al., 2000; Liu & Layland, 2002).

A Figura 3.5 ilustra o escalonamento de duas tarefas A e B utilizando-se a política EDF. As características das tarefas são definidas pelo conjunto (Tempo de Computação, Período e *Deadline*). A tarefa $A \supset (10, 20, 20)$ e a tarefa $B \supset (25, 50, 50)$.

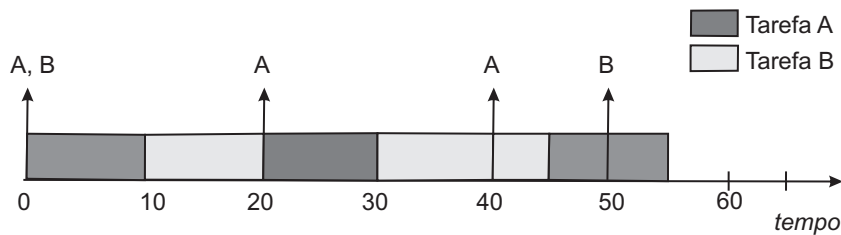


Figura 3.5: Exemplo da Utilização do Algoritmo de Escalonamento EDF. Adaptada de Peixoto (2008).

Na Figura 3.5, ao término de execução da tarefa A , no instante $t = 10$, B inicia seu processamento, porém, é interrompido por A no instante $t = 20$, visto que os *deadlines* de A e B , nesse instante, são $d_A = 30$ e $d_B = 50$, respectivamente. Novamente ao término de execução da tarefa A , no instante $t = 30$, B continua seu processamento (15u.t. restantes). No instante $t = 40$, a tarefa A não interrompe a tarefa B , visto que A e B apresentam os *deadlines* $d_A = 60$ e $d_B = 50$, respectivamente. Sendo assim, B termina seu processamento (5u.t. restantes) no instante $t = 45$. Nesse instante, a tarefa A inicia seu processamento. Embora o período de B tenha início em $t = 50$, B não interrompe A , visto que apresenta *deadline* maior ($d_A = 60$ e $d_B = 95$).

Algoritmo EBS - *Exigency Based Scheduling*

A política EBS (Casagrande et al., 2007), a qual realiza o escalonamento de requisições *Web* em sistemas *Soft-RT* não-determinístico, conforme apresentado na Seção 2.3, provê garantias de QoS absoluta em nível de aplicação.

Para avaliação da eficiência e do desempenho da política EBS, foi utilizada como abordagem, a modelagem e simulação orientada a eventos (Seção 4.2.1). Sendo assim, utilizando-se os conceitos de redes de filas (Seção 4.2.2), construiu-se um modelo que representa um servidor *Web* com qualidade de serviço, o qual descreve, portanto, os principais eventos que ocorrem em um sistema real do gênero, tais como: eventos para tratar chegada de requisições, solicitação de serviço e liberação de recurso, além de, oferecer

suporte à QoS. A Figura 3.6 ilustra tal modelo.

O servidor modelado é do tipo monoprocessado com uma fila única de espera para processamento. Nesse modelo foram implementados, além da EBS, os algoritmos de escalonamento FIFO, SJF e EDF, apresentados na Seção 3.3.2, de forma a comparar o desempenho de cada uma dessas políticas.

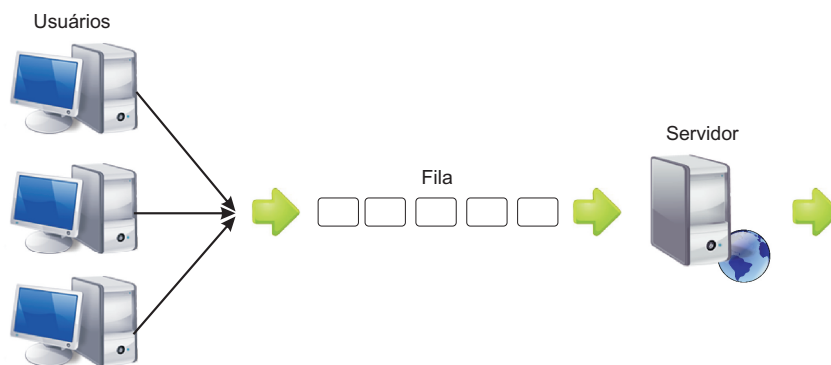


Figura 3.6: Representação do Modelo de Servidor *Web* Sequencial com QoS. Adaptada de Casagrande (2007).

O nível de desempenho, oferecido aos sistemas, especifica a qualidade de serviço oferecida ao mesmo. Para quantificar tal desempenho, uma métrica geralmente utilizada é a média do tempo de resposta (ART) do sistema que representa o tempo médio de residência das requisições de um usuário no sistema, ou seja, o intervalo entre a submissão e o completo recebimento do resultado da requisição (Jain, 1991).

Tal métrica é considerada, portanto, uma boa alternativa, visto que em seu cálculo estão inclusos os tempos em fila das requisições e, a manipulação desses tempos, de acordo com o parâmetro contratual de cada classe, influencia diretamente na melhora do desempenho de aplicações *Web*. Sendo assim, as requisições que estão mais próximas ou foram descumpridas receberão maior prioridade do escalonador, ao contrário daquelas que tole-ram maiores tempos em fila.

Portanto, no modelo proposto, a métrica escolhida é o **limite superior** para a média do tempo de resposta a ser garantido às requisições de um determinado usuário. Essa métrica, denominado, neste texto, **tempo médio de resposta contratado** pela *i-ésima* classe (C_i), é especificada previamente por um acordo entre o provedor de serviços e o usuário, e utilizado pelo escalonador como base para atribuição de prioridades. Para tanto, o valor da métrica, deve ser observado pelo servidor durante uma sessão.

A política desenvolvida não é preemptiva. Portanto, as novas requisições que chegam ao sistema e não encontram servidor disponível são transferidos para a fila de espera (Figura 3.6). Ao término da execução de uma requisição j do usuário u , o valor de t_{E_u} , média instantânea de tempo de resposta efetivamente oferecida ao usuário u , é recalculado. A Equação 3.1 mostra esse cálculo, o qual corresponde à média entre o antigo tempo de

resposta efetivo de u (t'_{E_u}) e o tempo de residência da requisição j recém atendida. Os valores de $time()$, $timeStamp_j$ e R_u representam, respectivamente, o tempo atual, o tempo de chegada da requisição e o número de requisições anteriormente submetidas por u .

$$t_{E_u} = \frac{(t'_{E_u} \cdot R_u) + (time() - timeStamp_j)}{R_u + 1} \quad (3.1)$$

Uma tabela de contratos é mantida atualizada pelo sistema. Cada entrada da tabela, correspondente a um usuário, relaciona o tempo médio de resposta contratado fixo, C_i , e o tempo médio efetivo t_{E_u} . A cada ciclo de escalonamento, a diferença, $C_i - t_{E_u}$, é comparada, conferindo maior prioridade à requisição do usuário cujo contrato esteja mais próximo de ser violado. Dessa forma, busca-se na fila de espera, a requisição mais urgente, ou seja, aquela que, dentre todas as existentes no sistema, apresenta o menor valor de **tempo de espera máximo**, que representa o quanto uma requisição ainda pode esperar na fila antes de começar a descumprir seu contrato³. Tal requisição, portanto, tem acesso ao servidor.

Para obter o tempo de espera máximo (D_j), também denominado *deadline*, aceitável pelo usuário u , antes que o valor do seu tempo médio de resposta efetivo (t_{E_u}) ultrapasse o seu valor de tempo médio de resposta contratado (C_i), isola-se a variável D_j , na Inequação 3.2, em que t_{W_j} expressa o tempo de espera em fila da requisição j até o momento.

$$\frac{(t_{E_u} \cdot R_u) + t_{W_j} + D_j}{R_u + 1} \leq C_i \quad (3.2)$$

A medida que o valor de D_j diminui, maior é a sua prioridade, pois maior é sua urgência. Em certas circunstâncias, requisições urgentes podem assumir os mesmos valores de *deadline*. Embora apresentem a mesma urgência, as requisições podem impor pesos distintos ao sistema, uma vez que, o **custo de processamento** é um fator impactante sobre a exigência imposta ao sistema. A minimização desse impacto torna-se, então, de extrema importância, visto que um sistema sob menor carga terá melhores condições para lidar com os requisitos de serviço de suas requisições.

Portanto, faz sentido também, dentre as mais urgentes, reordenar as requisições pelos seus valores esperados de processamento. Isso possibilita uma melhor utilização de recursos do sistema, como menores tamanhos de fila, visto que, requisições menores tendem a sair mais rapidamente do sistema, diminuindo o tempo de espera das demais (conforme apresentado na Seção 3.3.2), e conseqüentemente, afetando assim a quantidade de requisições aguardando por atendimento; e o que é mais importante, possibilita menores médias de tempo de resposta do sistema, contribuindo para obtenção de melhores níveis de qualidade de serviço.

Sendo assim, a atribuição de prioridades baseia-se tanto no *deadline* quanto no valor esperado do **tempo de processamento** da requisição. A Equação 3.3 mostra essa atri-

³Média do tempo de resposta daquele usuário vai descumprir o contrato.

buição de prioridades, em que a prioridade de uma dada requisição j em fila, do usuário u , é dada por P_j e as requisições que apresentarem maior urgência (menores valores de D_j) e menor custo esperado de processamento (t_{P_j}) serão classificadas como mais prioritárias.

$$P_j = D_j \cdot t_{P_j} = ((C_u \cdot (R_u + 1)) - (t_{E_u} \cdot R_u) - t_{W_j}) \cdot t_{P_j} \quad (3.3)$$

Em alguns casos o *deadline* pode assumir valores negativos, representando que o tempo que a requisição pode aguardar na fila é menor que zero, ou seja, o contrato foi violado. Se existirem duas requisições (R_1 e R_2) mais urgentes, com seus respectivos *deadlines* (D_i) negativos e tempos de processamento (t_{P_i}), por exemplo, uma requisição R_1 com $D_1 = -1$ e $t_{P_1} = 7.5u.t.$ e uma requisição R_2 com $D_2 = -2$ e $t_{P_2} = 2.5u.t.$. Se o escalonamento proposto (multiplicar o *deadline* pelo tempo de processamento) for aplicado nesse exemplo, embora R_2 seja mais urgente, a requisição R_1 receberia maior prioridade e seria inicialmente escalonada, devido $P_1 < P_2$ ($P_1 = -7.5u.t.$ e $P_2 = -5.0u.t.$). Nesses casos, é realizada uma correção, de forma a manter o objetivo proposto.

A Equação 3.4 representa tal correção, em que a prioridade das requisições mais urgentes, que ainda não tiveram seus *deadlines* descumpridos, é diretamente proporcional ao seu *deadline* e ao seu custo esperado de processamento. Já para aquelas com descumprimento de *deadline*, a prioridade é inversamente proporcional ao seu custo esperado de processamento, visto que, quanto menor for esse custo, menor será o valor de P_j resultante e portanto maior será sua prioridade de escalonamento. Sendo assim, garante-se que as requisições mais urgentes, independente de terem descumprido ou não seus *deadlines*, e com menores custos esperados de processamento sejam escalonadas primeiro.

$$P_j = \begin{cases} D_j \cdot t_{P_j} & \text{se } D_j \geq 0 \\ D_j \cdot \frac{1}{t_{P_j}} & \text{se } D_j < 0 \end{cases} \quad (3.4)$$

Dessa forma, foi possível a obtenção de bons resultados de confiabilidade, na qualidade de serviço oferecida, além de melhor desempenho e significativa diferenciação de serviço e, conseqüentemente um considerável aumento da satisfação contratual dos usuários do sistema como um todo. Tais resultados foram obtidos pela simulação do modelo especificado e podem ser observados nos cenários apresentados em Casagrande (2007).

Para execução dos experimentos de simulação, foram especificadas duas classes de serviço, uma classe A com um contrato mais estrito (valor de contrato menor) e uma outra classe B com um contrato mais “relaxado” (valor de contrato maior).

Cenários foram definidos, por meio da variação de alguns parâmetros, para análise de quais fatores a política de escalonamento EBS se apresenta ou não adequada.

Um dos parâmetros que define um cenário é a **porcentagem de variação de contrato** (P), ou seja, a porcentagem de discrepância dos serviços, oferecidos por cada algoritmo em relação ao tempo de residência em um sistema de escalonamento convencional

FIFO (*FIFO*). Utilizou-se, como base, médias de tempo de resposta que seriam oferecidas por um servidor convencional sem suporte à QoS, com o intuito de definir contratos de serviços viáveis e não contratos com tempos impossíveis de serem garantidos com qualquer tipo de algoritmo de escalonamento.

Foram atribuídos à P os valores: 5%, 10%, 20%, 30%, 40% e 50%. Sendo que, quanto maior for a variação dos contratos em relação ao escalonamento FIFO, maior e menor será a dificuldade de atendimento dos usuários das classes A e B , respectivamente. As Equações 3.5 e 3.6 ilustram os cálculos dos contratos de A e de B , respectivamente.

$$C_A = FIFO - (FIFO \cdot P) \quad (3.5)$$

$$C_B = FIFO + (FIFO \cdot P) \quad (3.6)$$

Outro parâmetro considerado é a **proporção de requisições** da classe A no sistema. Esse parâmetro especifica três situações: a primeira com um sistema sobrecarregado de requisições da classe que apresenta valor de contrato maior (10%A-90%B); a segunda com um sistema sobrecarregado com requisições da classe que apresenta valor de contrato menor (90%A-10%B); e a terceira, uma situação de igualdade no número de requisições de ambas as classes (50%A-50%B).

Fixou-se como 100.000, o número de requisições submetidas em cada cenário simulado, de forma a apresentar uma ampla amostragem dos dados. Além disso, com o intuito de obter confiabilidade estatística, realizou-se 30 vezes os experimentos (Bhattacharyya & Johnson, 1977), simulando os 18 cenários em cada uma delas.

As Figuras 3.7 a 3.9 ilustram a seleção de alguns cenários principais de variação de contrato (5% e 25%) e proporção de requisições (10%A-90%B, 50%A-50%B e 90%A-10%B), dentre todos os estudados, de forma a analisar a influência destes fatores sobre a eficácia dos algoritmos em respeitar os tempos de resposta estabelecidos nos contratos de cada classe. Nesses cenários, é realizado um comparativo entre o comportamento da política EBS com o de uma política sem suporte à QoS (FIFO). Vale lembrar que, além da política FIFO, foi realizado um comparativo entre as políticas EBS, SJF e EDF.

Nos gráficos das Figuras 3.7 a 3.9 são ilustradas as médias do tempo de resposta do sistema oferecidas ao longo do tempo para as classes de serviço com maior e menor dificuldade de atendimento, classes A e B , respectivamente. O eixo das abscissas informa o término de atendimento de uma requisição e o momento em que a média do tempo de resposta oferecida àquele usuário é atualizada. O eixo das ordenadas representam tais médias. Os valores dos contratos que definem as classes de serviço (A e B) são representados pelas retas horizontais C_A e C_B , respectivamente.

No cenário, apresentado pela Figura 3.7, há uma proporção de 10% de requisições da classe A e 90% de requisições da classe B . Na Figura 3.8, as proporções de requisições

de ambas as classes são iguais (50%A-50%B). Já a Figura 3.9 ilustra um cenário com predomínio de 90% de requisições da classe A e 10% de requisições da classe B.

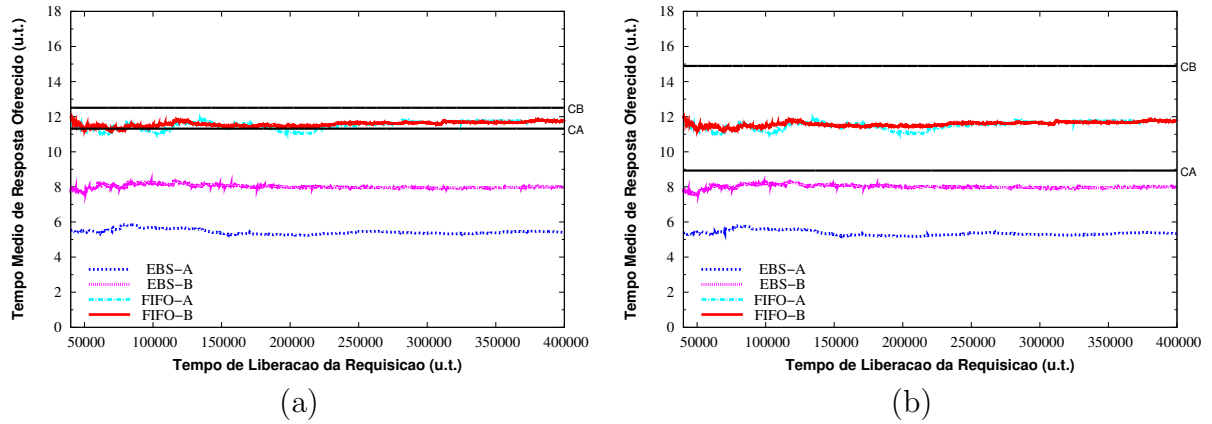


Figura 3.7: Representação do cenário de 10% de requisições A e 90% de requisições B, apresentados pelas políticas EBS e FIFO: (a) com 5% de variação de contrato. (b) com 25% de variação de contrato.

Conforme pode ser observado pelas Figuras 3.7 a 3.9, simulações do modelo utilizado em Casagrande (2007), sob cenários de interesse, produziram resultados positivos. Todavia, embora a meta, de manter o tempo médio de resposta abaixo do limite estipulado em contrato tenha sido atingida nos experimentos realizados, considerações acerca da aplicabilidade do método, em situações práticas, suscitam conjecturas sobre as condições em que contratos de QoS dessa natureza sejam suficientes para as necessidades ou expectativas dos usuários da aplicação *Web*, posto que a política EBS objetiva apenas QoS absoluta.

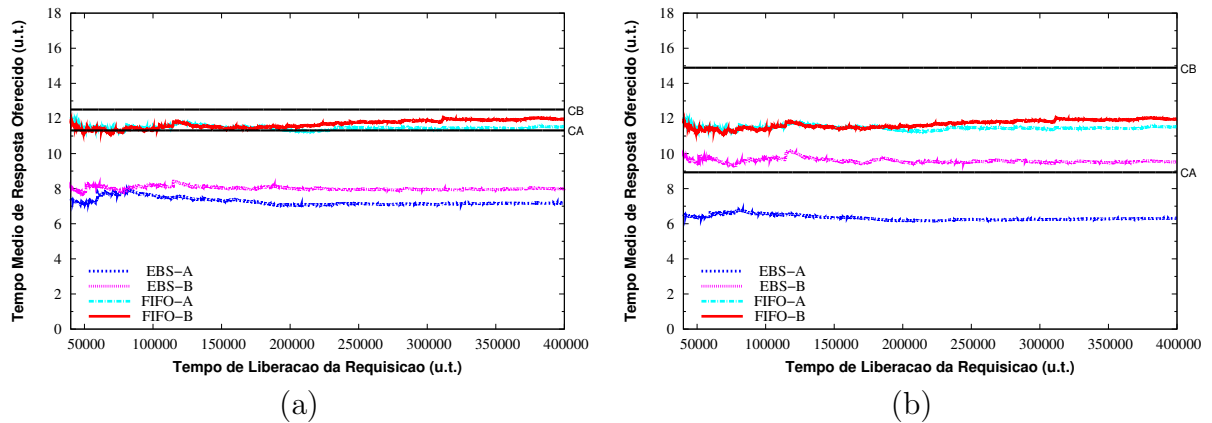


Figura 3.8: Representação do cenário de 50% de requisições A e 50% de requisições B, apresentados pelas políticas EBS e FIFO: (a) com 5% de variação de contrato. (b) com 25% de variação de contrato.

Pode-se observar, pela Figura 3.9, que mesmo com menor exigência contratual, um usuário B obtém melhores valores médios de tempo de resposta do sistema, o que denomina-se uma *inversão de prioridade* em termos de QoS relativa.

Ressalte-se, em particular, que essa “inversão” não apresenta problemas com relação

a QoS absoluta. Porém, em termos de modelo de negócios, se os usuários estiverem acostumados com contratos de qualidade de serviço baseados em prioridade, pode haver certo desconforto na sua forma de percepção de QoS.

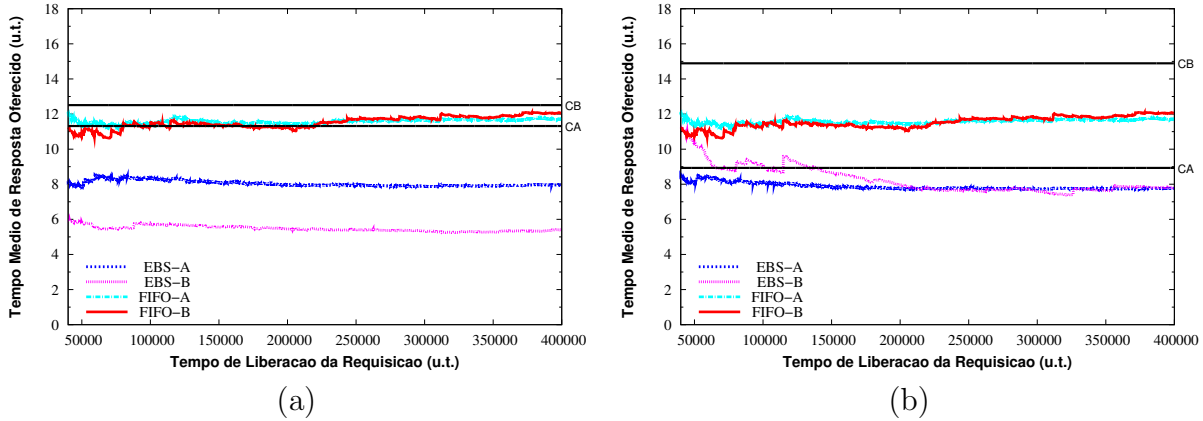


Figura 3.9: Representação do cenário de 90% de requisições A e 10% de requisições B, apresentados pelas políticas EBS e FIFO: (a) com 5% de variação de contrato. (b) com 25% de variação de contrato.

Sendo assim, seria conveniente dispor de um contrato que contemplasse QoS absoluta e relativa em uma mesma política, garantindo, dessa forma, um máximo tempo médio de resposta para cada classe, como também um melhor atendimento a uma classe cujo tempo médio de resposta contratado seja inferior. Este é o objetivo do presente projeto de pesquisa.

3.4 Considerações Finais

O presente capítulo introduziu os principais fundamentos da área de **sistemas de tempo real**, os quais são responsáveis por estabelecer parâmetros de serviço em termos temporais. Conceitos básicos são discutidos (Seção 3.2), apresentando suas três vertentes, **Hard-RT**, **Soft-RT** e **Firm-RT** (Seção 3.2.1).

Além dos conceitos primordiais ao entendimento da área, o capítulo apresenta o **escalonamento em sistemas RT**, incluindo algumas formas de classificação para o mesmo, bem como alguns **algoritmos clássicos** utilizados na literatura, tais como: **FIFO**, **SJF**, **EDF**, além da política **EBS**.

Uma das formas para realizar a validação de **algoritmos de escalonamento** é a utilização do modelamento baseado em **simulação de sistemas**, que oferece facilidades para a representação do comportamento de um sistema do mundo real sobre o tempo e o qual será abordado no capítulo 4 subsequente.

Simulação de Sistemas

4.1 Considerações Iniciais

Como alternativa a métodos analíticos, ferramentas úteis, como a **simulação de sistemas** têm sido extensivamente utilizadas para o desenvolvimento e análise de novos algoritmos de escalonamento.

A simulação consiste no estudo do comportamento e reações de um determinado sistema, de forma a reproduzir suas propriedades e comportamentos, possibilitando a manipulação e o estudo detalhado desse sistema (Banks et al., 2000; MacDougall, 1987; Shannon, 1998).

A representação de comportamento é realizada por meio de um modelo construído a partir de suposições operacionais de um sistema real que, após validado, pode ser utilizado para investigar questões do tipo “o que aconteceria se” a cerca de um sistema real. Podendo, dessa forma, ser considerado como uma descrição do sistema real.

A solução desse modelo pode ser realizada, por meio analítico, utilizando a lógica dedutiva da matemática, ou por meio de simulação, empregando métodos numéricos na análise (Law & Kelton, 2006).

A solução por simulação é útil quando o estudo por experimentação torna-se inviável, dada a complexidade do sistema real; quando se pretende analisar isoladamente a influência de mudanças organizacionais ou ambientais no comportamento de um sistema; ou ainda para testar novos projetos ou políticas antes de serem efetivamente implementados (Banks et al., 2000).

A execução de modelos de simulação possibilita o fornecimento de resultados mais pre-

cisos sem a necessidade de se interferir no sistema real. Tais resultados, quando analisados estatisticamente, produzem informações que podem contribuir na tomada de decisões que visam a solução de problemas. Essa será a técnica utilizada neste projeto de pesquisa.

4.2 Simulação de Sistemas

Para estruturação e análise do modelo de simulação é necessária a definição de alguns conceitos e nomenclaturas. Dessa forma, **estado** de um sistema é definido pelo conjunto de variáveis necessárias para descrevê-lo em um determinado período. Variáveis de estado, em um sistema bancário, por exemplo, podem ser representados pelo tamanho da fila de *jobs*. De acordo com o tipo de estados predominantes no sistema ou no foco da análise, o sistema pode ser classificado como **discreto** ou **contínuo**.

Em **sistemas discretos** (Freitas, 2008), o estado sofre mudanças instantâneas em pontos distintos de tempo, como por exemplo, o número de clientes em uma fila. Já em **sistemas contínuos**, mudanças ocorrem de forma contínua ao longo do tempo, como o volume de água em uma hidrelétrica, o que possibilita a determinação dos valores das variáveis de estado a qualquer instante.

Os modelos de simulação ainda podem ser classificados em: (a) **estáticos** ou **dinâmicos** - **modelos estáticos** caracterizam-se por representar o estado de um sistema em um instante, sendo que, em suas formulações a variável tempo não é considerada. Já os **modelos dinâmicos** caracterizam-se por representar as alterações de estado do sistema ao longo da contagem de tempo de simulação. (b) **determinísticos** ou **estocásticos** - **modelos determinísticos**, em suas formulações, não utilizam componentes probabilísticos (variáveis aleatórias). Dado um conjunto fixo de valores de entrada, o resultado da saída será sempre o mesmo quando a simulação for executada. Já os **modelos estocásticos** fazem uso dessas variáveis. Os valores dos parâmetros de saída devem variar dependendo da variação estocástica dos valores de entrada intermediárias (internos) do modelo.

Sendo assim, pode-se melhor formalizar o tipo de simulação utilizado neste trabalho como simulação baseada em **eventos discretos**, com alterações de estado ao longo do tempo (**dinâmico**) e com parâmetros **estocásticos**.

Além da descrição da estrutura estática de um sistema, a modelagem de um sistema abrange também a representação de sua composição dinâmica, ou seja, o modo que o sistema realiza trabalho, podendo ser descrita em termos de entidades, como: **atividades**, **processos** e **eventos** (MacDougall, 1987).

Uma **atividade** é a menor unidade de trabalho na visão que se tem do sistema. Cada atividade tem um tempo de execução associada a ela. O conjunto de atividades relacionadas logicamente constituem um **processo**. O tempo de execução de um processo (ignorando concorrência) é formado pela soma das execuções e tempos de atrasos de cada uma dessas atividades. A inicialização das atividades é ativada pelos **eventos**. Um evento

representa uma mudança de estado de alguma entidade do sistema, como por exemplo, chegada de um *job*, início de uma nova execução, parada de um *job* ou uma CPU que passa do estado ocioso para o estado ocupado. Essa mudança de estado resulta da ação de uma atividade.

4.2.1 Linguagens de Simulação

Para o desenvolvimento e a solução de modelos de simulação, linguagens de simulação têm sido desenvolvidas e utilizadas nas mais diversas áreas, conforme pode ser observado em Cabral & Souto (2004); Martin (1997); Posse & Vangheluwe (2007). Como abordado (Seção 4.2), as entidades (atividade, processo e evento) descritas, são utilizadas para estruturar o modelo de simulação a ser construído. De acordo com a entidade que embasa a descrição do sistema, as linguagens de simulação podem ser classificadas em: **orientados à atividade**, a **evento** ou a **processo**, sendo as orientadas a evento e a processo, as mais utilizadas.

Linguagens **orientadas a processo**, como CSIM (Hlavicka & Racek, 2002; Schwetman, 1986), SIMULA (Birtwhistle et al., 1979) e ASPOL (MacDougall & McAlpine, 1973) são recomendadas para implementar modelos de simulação de grande escala. Programas de simulação escritos nessas linguagens podem ser construídos como uma descrição direta do sistema em questão. Essa similaridade entre o modelo e o sistema permite assegurar que tal modelo seja uma representação válida do sistema, principalmente em um ambiente de desenvolvimento no qual o projeto do sistema sofre constantes mudanças.

Nas linguagens de simulação **orientadas a eventos**, como SIMPACK (Fishwick, 1992) e SMPL (MacDougall, 1987), o programa de simulação é organizado como um conjunto de rotinas ou seções de acontecimentos (MacDougall, 1987). Essas linguagens tendem a impor ao modelador uma visão global e de alto nível do sistema, agrupando ações de atividades logicamente não relacionadas em uma única rotina de evento, o que possibilita a perda da “identidade” com a estrutura do sistema e dificulta a modificação do mesmo. Além disso, essa abordagem permite que se conheça o estado de qualquer entidade do sistema em qualquer instante de tempo. Sendo assim, tais linguagens são mais adequadas a modelos de pequena e média escala (Cabral & Souto, 2004; Jain, 1991; Lazowska et al., 1984; MacDougall, 1987).

4.2.2 Simulação de Filas

Filas estão presentes em diversas situações no mundo real. A ocorrência de filas se dá à medida que a procura por um determinado serviço é maior que a capacidade do sistema de prover tal serviço. Na Figura 4.1 são ilustrados os elementos que compõem uma fila, bem como algumas características que fornecem uma descrição adequada de um sistema de filas.

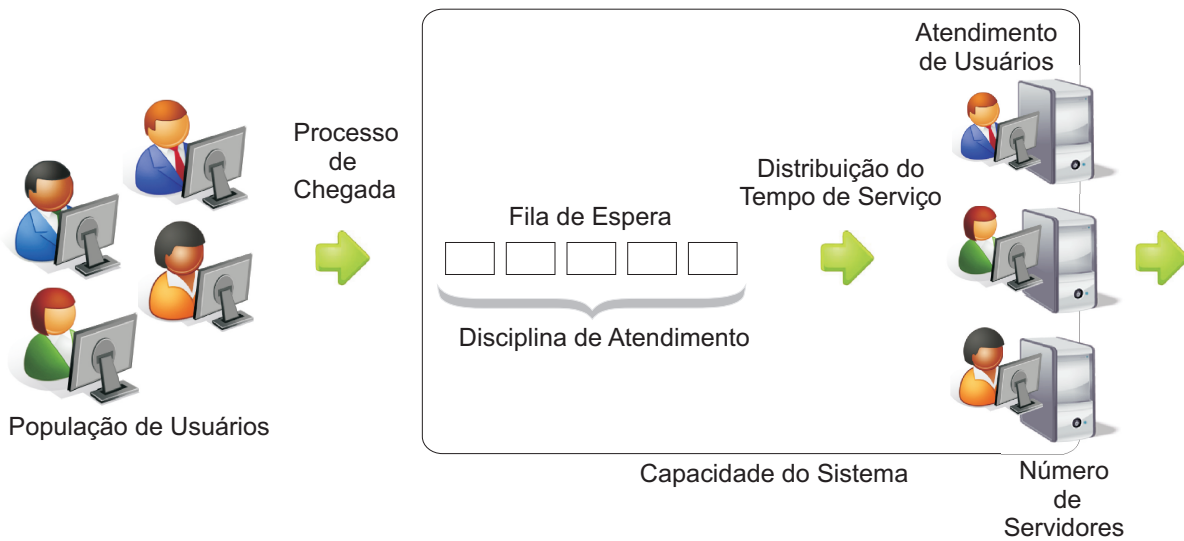


Figura 4.1: Ilustração dos Elementos e das Características de um Sistema de Fila. Adaptada de Prado (2004).

Pode-se observar pela Figura 4.1, que a partir de uma população, usuários formam uma fila, aguardando por um determinado tipo de serviço oferecido pelos servidores, bem como sua disponibilidade e, ao término do atendimento, os usuários saem do sistema. O conhecimento das características apresentadas são essenciais para a descrição adequada do sistema a ser modelado. Sendo assim, tais características são detalhadas a seguir:

- **Distribuição de chegada dos usuários:** os **processos de chegada** apresentam um comportamento estocástico, ou seja, desenvolvem-se no tempo e no espaço de acordo com as leis de probabilidade. Dessa forma, é fundamental o conhecimento da distribuição de probabilidade (ver Tabela 4.2), que descreve os tempos entre as sucessivas chegadas dos usuários. Caso os usuários cheguem nos instantes t_1, t_2, \dots, t_j , as variáveis aleatórias $t_j = t_j - (t_{j-1})$ são chamadas de intervalos entre chegadas;
- **Distribuição do tempo de serviço:** refere-se ao tempo de utilização dos serviços, dado por uma distribuição de probabilidade. Sendo que, a medida que o tempo de espera aumenta, o usuário opta pelo permanecimento ou não na fila, de acordo com a capacidade do servidor em atender as requisições. O controle de admissão pode ser utilizado para a identificação dessa distribuição de tempo de serviço e para a tomada de decisão, de forma que se a quantidade de requisições ultrapassar um limiar, o sistema as descarte;
- **Número de servidores:** são previamente definidos, determinando-se, assim, o número de estações de serviços paralelos que podem atender os usuários simultaneamente;

- **Capacidade do sistema:** corresponde à limitação física da quantidade de espaço na fila. Se a capacidade do sistema for grande, considera-se que a fila seja infinita, enquanto que se as filas alcançarem um determinado comprimento, de forma que nenhum usuário possa entrar no sistema até que haja um espaço disponível obtido por meio do atendimento de um usuário, considera-se, assim, um sistema de filas finitos, ou seja, existe um limite finito do tamanho máximo do sistema. A capacidade do sistema inclui, tanto os usuários esperando em fila como os que estão sendo atendidos;
- **Tamanho da população:** número potencial (finito ou infinito) de usuários que podem chegar a um sistema;
- **Disciplina de atendimento de filas:** determina a forma como os usuários são escolhidos para serem atendidos após a formação da fila. A disciplina mais utilizada é a FCFS (*First Come First Served*), ou seja, o primeiro a chegar é o primeiro a ser servido. Existem outras disciplinas, tais como, LCFS (*Last Come First Served*), ou seja, o mais recente é o primeiro a ser servido e, diversas outras baseadas em esquemas de prioridade. Existem duas situações gerais em disciplinas de prioridade: **preemptivo** e **não preemptivo**. No primeiro caso, o usuário com maior prioridade pode entrar em serviço independentemente dos demais usuários com menor prioridade estarem sendo servidos, esses usuários têm seus trabalhos interrompidos, podendo ser reiniciados posteriormente. Já no caso de disciplinas não preemptivas, usuários com uma maior prioridade vão para o início da fila, entrando em serviço ao término das requisições sendo servidas. Essas prioridades podem se basear em questões como: tempo esperado de atendimento, tamanho do trabalho gerado pelo usuário, sensibilidade de atrasos e qualidade de serviço.

4.2.2.1 Redes de Filas

Uma técnica amplamente utilizada para modelagem de sistemas com ocorrência de filas (servidores *Web*, por exemplo) é a **Rede de Filas**. A ocorrência de filas se dá à medida que a procura por um determinado serviço é maior que a capacidade do sistema de prover tal serviço. Sendo assim, técnicas de Redes de Filas, da **Teoria de Filas**, são utilizadas para a modelagem desses tipos de sistemas, a qual, por meio de análises matemáticas detalhadas, procuram tanto satisfazer os usuários quanto ser viável economicamente para o provedor do serviço.

Redes de Filas são representadas por entidades que oferecem serviços aos usuários do sistema, por meio de **centros de serviço**. Um centro de serviço consiste de um ou mais servidores (correspondentes aos recursos do sistema) e uma fila, de capacidade finita ou infinita, composta por um conjunto de usuários que aguardam pelo atendimento do

serviço. Um exemplo de recurso seria a CPU de um computador. A Figura 4.2 ilustra diferentes tipos de centros de serviços (Soares, 1992).

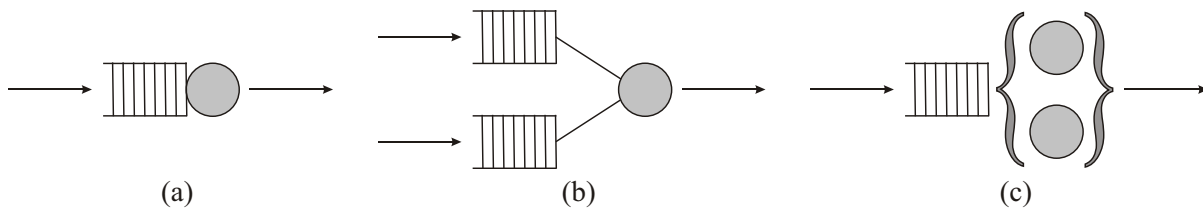


Figura 4.2: Exemplos de Centros de Serviços: (a) com 1 fila e 1 servidor; (b) com 2 filas e 1 servidor e (c) com 1 fila e 2 servidores. Adaptada de Soares (1992).

Baseado no comportamento de entrada e saída dos usuários, os modelos de Redes de Filas podem ser classificados em: **modelos abertos**, **fechados** ou **mistos**. **Modelos abertos** caracterizam-se pela entrada e, obrigatoriamente saída dos usuários no sistema. Em **modelos fechados**, o sistema é composto por um número fixo de usuários, de forma que não há entrada nem saída de usuários. Já **modelos mistos** correspondem a uma combinação dos dois anteriores, sendo aberto para determinadas classes de usuários e fechado para outras. A Figura 4.3 ilustra esses três modelos de Redes de Filas (Lazowska et al., 1984).

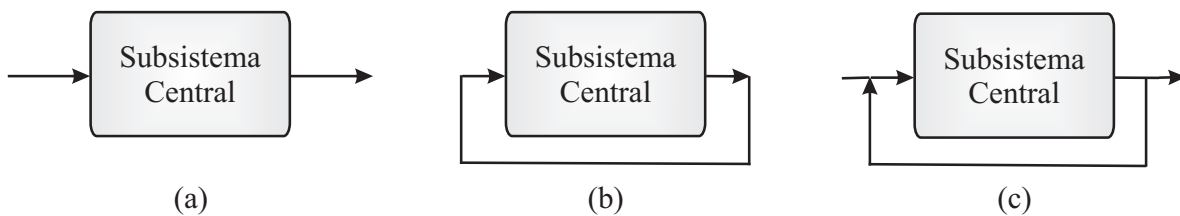


Figura 4.3: Ilustração dos Modelos de Redes de Filas: (a) Aberto; (b) Fechado e (c) Misto. Adaptada de Lazowska et al. (1984).

Para facilitar a descrição das características desses modelos, foram definidos padrões de representação, entre eles têm-se a notação de Kendall, representada pela tupla, conjunto de variáveis, $A/S/c/k/m/D$ e descritas na Tabela 4.1.

Tabela 4.1: Notação de Kendall

Símbolo	Descrição
A	Distribuição que descreve o intervalo de chegada dos usuários
S	Distribuição que descreve o tempo de serviço
c	Número de servidores
k	Capacidade do sistema (número máximo de usuários suportados pelo sistema)
m	Tamanho da população (finita ou infinita)
D	Disciplina de fila a ser utilizada

Os parâmetros k e m quando assumem valores infinitos, geralmente, são omitidos. O parâmetro D também é omitido quando representa a disciplina de fila FIFO. As distribuições de probabilidade utilizadas para descrever variáveis contínuas, como o tempo entre chegadas e o tempo de serviço, representadas pelas notações A e S , respectivamente, são descritas, normalmente, pelos símbolos ilustrados na Tabela 4.2.

Tabela 4.2: Descrição das Distribuições de Probabilidade

Símbolo	Descrição
D	Distribuição determinística (valores de intervalos de chegada e tempo de serviço constantes)
M	Distribuição exponencial
E_k	Distribuição Erlang de estágio k
H_k	Distribuição hiper-exponencial de estágio k
G	Distribuição arbitrária

Dentre as distribuições apresentadas, a exponencial, segundo Jain (1991), é amplamente utilizada em modelagem de filas.

A notação e a definição de algumas variáveis utilizadas para a caracterização dos componentes de redes de filas são apresentadas na Tabela 4.3.

Tabela 4.3: Notação e Definições Básicas dos Componentes de Redes de Filas.

Símbolo	Descrição
τ	tempo entre chegadas
λ	taxa média de chegada, definida por $\frac{1}{E[\tau]}$, onde $E[\tau]$ é o tempo médio entre chegadas
s	tempo de serviço por usuário
μ	taxa média de serviço por servidor, definida por $\frac{1}{E[s]}$, onde $E[s]$ é o tempo médio de serviço. A taxa de serviço total para c servidores é $c\mu$
n_u	número de usuários no sistema ou comprimento da fila, incluindo tanto os usuários que estão recebendo serviço quanto os que estão esperando na fila
n_q	número de usuários esperando para receber serviço. Sempre n_q é menor que n
n_s	número de usuários recebendo serviço
r	tempo de resposta ou tempo no sistema. Inclui tanto tempo de espera quanto tempo de recebimento de serviço
w	tempo de espera ou tempo de fila, ou seja, o intervalo de tempo entre a chegada e o instante em que inicia o serviço
T	período de tempo observado
A	número de chegadas de usuários no tempo T
C	número de serviços completados no tempo T
B	tempo em que o servidor permaneceu ocupado durante T

Na Tabela 4.3, as variáveis tempo de resposta (r) e número de usuários (n) no sistema merecem destaque, dado que algumas equações, para sistemas de fila única, podem ser consideradas. O tempo de resposta, tempo gasto pelo usuário no sistema, é dado pela soma do tempo de espera do usuário na fila e o tempo de serviço recebido por esse usuário, ou seja, $r = \omega + s$ (Jain, 1991). Já o número de usuários no sistema é dado pela soma dos usuários que estão na fila esperando pelo recebimento de serviço e aqueles que estão recebendo serviço, ou seja, $n_u = n_q + n_s$ (Jain, 1991).

Baseado na teoria de filas, pode-se realizar uma análise matemática de diversos processos, tais como: chegada em fila, espera em fila e oferecimento de serviço pelos servidores. Tal análise possibilita a derivação e o cálculo de várias medidas de desempenho, incluindo tempo de espera em fila ou no sistema, comprimento de fila, número de elementos recebendo serviço e a possibilidade de ocorrência de certos estados do sistema, como ocioso, ocupado, tendo algum servidor disponível ou ter que esperar certo instante de tempo para ser servido. Tais medidas de desempenho podem ser obtidas por meio de **relações algébricas**.

Por serem de grande relevância à avaliação de desempenho de modelos de Redes de Filas, algumas relações algébricas são denominadas **Leis Operacionais**. Essas Leis Operacionais são válidas tanto para sistemas de fila única quanto para sistemas com rede de várias filas.

4.3 Considerações Finais

Existem diversos métodos com o intuito de realizar a mensuração das propriedades de desempenho e confiabilidade de novas estratégias de escalonamento.

O presente capítulo introduziu os principais conceitos de **simulação de sistemas**, a qual tem como característica inerente a elaboração de um modelo de um sistema real (ou hipotético) e a condução de experimentos com a finalidade de entender o comportamento de um sistema ou avaliar sua operação.

Além dos conceitos primordiais ao entendimento de sistemas de simulação e suas peculiaridades, o presente capítulo, também apresenta uma técnica (Seção 4.2.2) e a teoria a partir da qual ela é fundamentada, para a modelagem de sistemas com ocorrência de **filas**, foco deste trabalho.

Integrando QoS Relativa e Absoluta

5.1 Considerações Iniciais

Um amplo leque de técnicas para provisão de QoS em servidores *Web*, baseados tanto na diferenciação de serviço quanto na garantia do cumprimento de requisitos temporais, encontram-se na literatura, porém nenhuma delas trata de maneira sistemática a relação de QoS relativa e absoluta em uma mesma técnica.

Neste capítulo é introduzida uma nova técnica que considera QoS relativa e absoluta e, portanto, lida com o problema da ocorrência de inversão de prioridade apresentada pela política EBS, conforme descrito na Seção 1.2, em que usuários mais exigentes recebem um atendimento pior (em um maior tempo médio de resposta) em relação a usuários menos exigentes. É também apresentada uma descrição detalhada dos experimentos desenvolvidos, bem como explicitados os resultados obtidos com a técnica proposta.

O método aplica a técnica de realimentação adaptativa e contratos virtuais para ajustar em tempo de execução os parâmetros de operação do Algoritmo EBS, afim de evitar inversões de prioridades cuja operação seja associada a um contrato de QoS com garantias absolutas e relativas de limite superior de tempo médio de resposta.

Nas Seções subsequentes, serão explicitados e introduzidos os conceitos relacionados para realização da proposta do novo método, bem como o método em si.

5.1.1 Conceitos Relacionados

Garantir que o tempo médio entre eventos de um dado processo seja superiormente limitado dentro de uma janela de ocorrências conveniente pode ser útil em diversas formas. Pode ser de ajuda para medir o *throughput* de sistemas estocásticos e para estimar demandas de recursos. Para sistemas computacionais interativos, isso representa não apenas um parâmetro de QoS praticável face ao não determinismo do ambiente, mas também uma métrica temporal de responsividade com significado intuitivo para a percepção do usuário, tendo assim um impacto relevante na qualidade de serviço (Henriksson et al., 2004), (Kang et al., 2003), (Lu et al., 2001), (Eggert & Heidemann, 1999).

Mesmo em tais sistemas o projeto de algoritmos de escalonamento de tarefas que competem pelos recursos disponíveis não é simples. As dificuldades para abordagens analíticas direcionam interesses para métodos heurísticos. Ainda assim, em ambientes com pouca previsibilidade, algoritmos de escalonamento baseados no pré-ajuste de parâmetros fixos em fase de projeto tende a ser ineficiente em termos de demandas calculadas com base no pior caso, ou ineficazes frente às variações na carga ou na capacidade de processamento.

Um conceito de crescente relevância na área de sistemas RT é o de *Feedback Scheduling* (escalonamento retroalimentado), que corresponde à aplicação dos fundamentos da Teoria de Controle em problemas de escalonamento. A abordagem baseia-se no princípio da retroalimentação negativa, em que a saída do sistema é comparada a um valor de referência desejado, e a diferença entre eles é tomada como entrada para o sistema, e manipulada de modo a causar neste uma reação contrária ao desvio entre referência e saída. Com isso é possível controlar o mecanismo de escalonamento em tempo de operação, de maneira a compensar variações na carga ou nos parâmetros do sistema. *Feedback Scheduling* constitui um dos paradigmas mais promissores dentro da área de escalonamento RT em ambientes não determinísticos (Sha et al., 2004), (Lu et al., 2001).

5.2 Método Proposto

Para resolver o problema de inversão, o método proposto considera a existência de contratos virtuais adaptativos vc_{EBS} (Figura 5.1), de forma a priorizar ainda mais as restrições dos usuários mais exigentes e a relaxar ainda mais as restrições dos usuários menos exigentes, desde que os tempos médios de resposta oferecidos a esses usuários sejam mantidos abaixo de seus respectivos limites superiores contratados (C_A e C_B). Isso é possível, visto que, os tempos médios de resposta efetivamente oferecidos aos usuários da classe de serviço B são bem menores que os estabelecidos em cada um de seus contratos, conforme pode ser observado pela Figura 5.1.

Essa solução possibilita ao escalonador priorizar uma requisição (classe A) cuja QoS esteja próxima ao limite especificado, em detrimento de outra (classe B), cujo serviço,

fornevido ao longo do tempo, tenha sido eventualmente realizado com qualidade muito superior a do nível contratado. Dessa forma, embora obtenha-se um aumento nos tempos médios de resposta oferecidos aos usuários menos exigentes (classe *B*), consegue-se diminuir os tempos médios de resposta oferecidos aos usuários da classe mais exigente (classe *A*), revertendo, portanto, o cenário de inversão de prioridade apresentado pela política EBS.

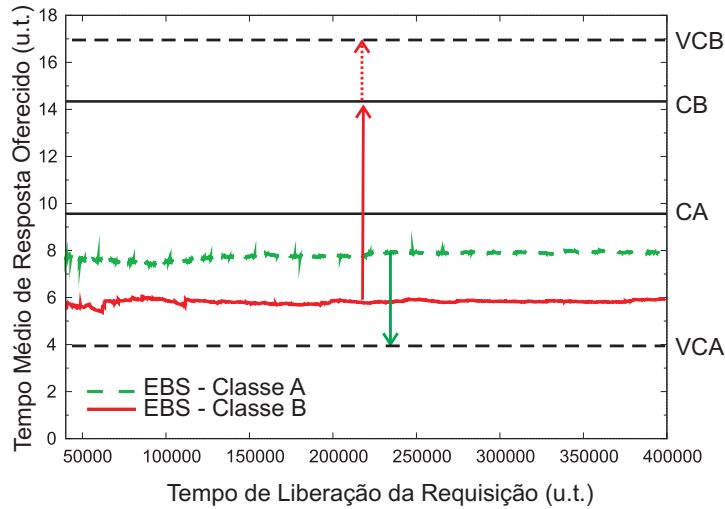


Figura 5.1: Representação do cenário de ocorrência de inversão de prioridade em termos de QoS relativa e solução considerando-se os contratos virtuais vc_A e vc_B .

O método proposto consiste primeiramente na análise e investigação dos cenários em que pode ser observada a ocorrência da inversão de prioridade. Na verdade, os contratos virtuais (vc_{EBS}) passam a atuar antes mesmo da inversão ocorrer. Portanto, a medida em que os tempos médios de resposta oferecidos aos usuários da classe *B* diminuem e se aproximam dos tempos médios de resposta oferecidos aos usuários da classe *A*, devem haver mecanismos adaptativos para ajustar os valores dos contratos virtuais em função da distância desses tempos médios de resposta. O Algoritmo 5.1 descreve o cálculo do valor do contrato virtual vc_{EBS} .

O contrato virtual (vc_{EBS}) atua mais intensamente nos cenários em que o tempo médio de resposta (ART) oferecido aos usuários da classe *B* são menores e/ou bastante próximo ao ART oferecido aos usuários da classe *A*. Nesses cenários, para garantir um atendimento melhor aos usuários da classe *A* e não permitir que a inversão de prioridade ocorra em nenhum cenário e ao longo de toda a simulação, um limiar t (*threshold*), que define o limite inferior para os ARTs oferecidos à classe *B*, é estabelecido. Este limiar (Equação 5.1) é definido como sendo o valor médio entre o contrato da classe *A* e o ART oferecido aos usuários dessa classe.

$$t = ((C_A + t_{E_A})/2) \quad (5.1)$$

Algoritmo 5.1: para calcular o valor dos contratos virtuais vc_{EBS} .

```

1   $k \leftarrow ((C_A + C_B)/2)$ ;
2  begin
3  | se  $tE_B < t$  então
4  |   | se ( $classe = A$ ) então
5  |   |   |  $vc_{EBS} = C_A - k$  ;
6  |   |   | senão
7  |   |   |   |  $vc_{EBS} = C_B + (k * x)$  ;
8  |   |   |   | fim
9  |   | senão
10 |   |   | se ( $classe = A$ ) então
11 |   |   |   |  $k \leftarrow 0$  ;
12 |   |   |   |  $vc_{EBS} = C_A - k$  ;
13 |   |   |   | senão
14 |   |   |   |   |  $vc_{EBS} = C_B + k$  ;
15 |   |   |   |   | fim
16 |   | fim
17 end

```

Sendo assim, a medida em que os ARTs oferecidos aos usuários da classe B diminuem e se aproximam do limiar t , o valor do contrato virtual é atualizado de forma a priorizar ainda mais as restrições dos usuários da classe A e relaxar ainda mais as restrições dos usuários da classe B . Para tanto, em se tratando de uma requisição de um usuário da classe A , o valor do contrato a ser considerado pelo sistema não é mais o seu contrato real C_A , mas o valor do contrato real C_A subtraído de uma certa constante k .

Já para as requisições dos usuários da classe B , o sistema passa a considerar um contrato virtual cujo valor corresponde ao valor contratado real (C_B) pelos usuários dessa classe adicionado à mesma constante k , porém é dado um peso maior à essa constante, multiplicando um parâmetro (x) de entrada do sistema. Esse parâmetro é empírico e pode ser modificado de forma a ser ajustado auto-adaptativamente pelo próprio sistema. Isso é necessário pois observou-se que os ARTs oferecidos aos usuários dessa classe foram bem inferiores aos valores estabelecidos em seus contratos (C_B), conforme pode ser observado pela Figura 5.1. O valor da constante k considerada corresponde à média dos valores contratados (C_A e C_B) pelos usuários das classes A e B , respectivamente.

O contrato virtual, conforme mencionado, atua também em cenários em que não ocorre a inversão de prioridade, porém, de forma menos intensa, atuando somente em requisições dos usuários da classe B . O sistema considera o contrato virtual como sendo o valor do contrato real estabelecido com o usuário adicionado ao valor da constante k . Ressaltando que isso foi necessário devido aos usuários da classe B estarem recebendo uma QoS superior ao que foi estabelecido em seu contrato.

O funcionamento passo-a-passo do método proposto é ilustrado na Figura 5.2.

Considera-se um conjunto de usuários, representando um cenário em que as inversões ocorrem, o qual é um cenário em que há um número maior de usuários mais exigentes (classe A). As requisições (classes A e B) que chegam e não encontram servidor disponível, aguardam na fila de requisições. A medida que o servidor encontra-se disponível, as requisições dos usuários são atendidas e estes obtêm seus ARTs (t_{EA} e t_{EB}). Caso os ARTs oferecidos a um usuário menos exigente (classe B), seja menor que os oferecidos aos mais exigentes, trata-se de uma situação de inversão de prioridade. Sendo assim, o método proposto passa a atuar, relaxando-se as restrições temporais dos usuários menos exigentes e priorizando ainda mais as requisições dos mais exigentes, diminui-se, portanto, os ARTs oferecidos a esses usuários e reverte-se, assim, o problema de inversão de prioridade. De acordo com a Figura 5.2, o contrato virtual (vc_{EBS}) é baseado nos princípios de escalonamento realimentado, em que a cada ciclo de escalonamento o valor do contrato é recalculado.

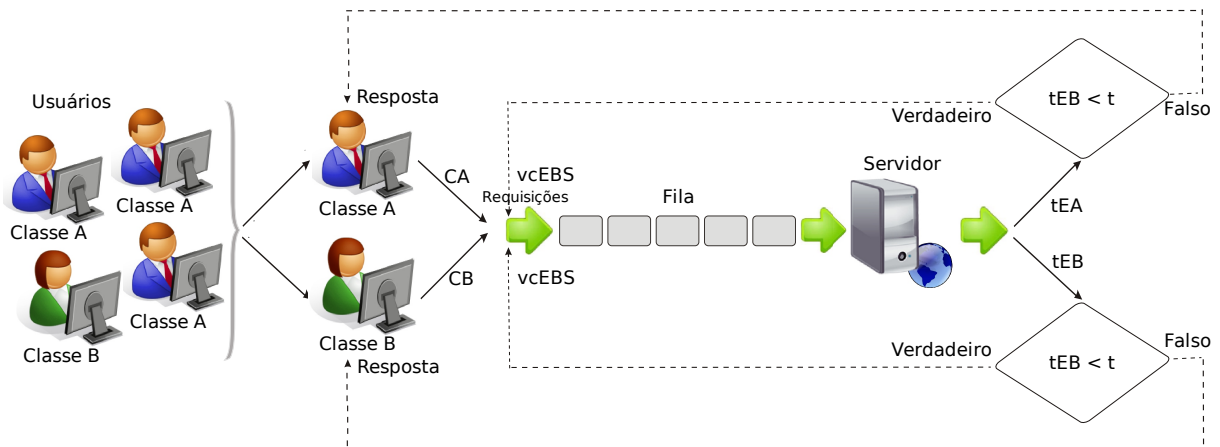


Figura 5.2: Passos do método proposto.

5.2.1 Metodologia

Para avaliar a eficiência e o desempenho do método proposto foi utilizada a modelagem e simulação orientada a eventos. A simulação possibilita uma maior flexibilidade para o estudo, bem como a obtenção de melhores estimativas de comportamento do sistema e previsões de desempenho após modificações. Além disso, a partir dos resultados obtidos, modelos mais completos e representativos do problema real podem ser construídos. Sendo assim, o mesmo modelo baseado em fila considerado em Casagrande et al. (2007) foi utilizado para representar um servidor *Web* com suporte à QoS (Figura 3.6).

Além disso, foi utilizado também o mesmo pacote de simulação SimpackJ, o qual oferece um conjunto de ferramentas para simulação orientada a eventos. O pacote SimpackJ é constituído por uma ampla gama de métodos voltados à simulação de redes de filas (Kumar & Majhi, 2004). Esse pacote foi desenvolvido a partir da biblioteca SMPL (Mac-

Dougall, 1987), com o acréscimo de algumas rotinas e a modificação do paradigma de programação para a orientação a objetos.

A escolha dessa ferramenta de simulação para o desenvolvimento do projeto deve-se à experiência na utilização do mesmo em trabalhos anteriores no GSDPC (Grupo de Sistemas Distribuídos e Programação Concorrente), o que facilitou o aprendizado. Além disso, como o seu código fonte é disponível livremente, foi possível modificá-lo quando necessário, característica muito importante para sua escolha, pois modificações foram necessários para dar suporte às requisições baseadas em tempo real bem como às outras características, em trabalhos anteriores.

5.3 Experimentos

Essa seção apresenta os resultados experimentais da utilização do método proposto, baseado em contratos virtuais adaptativos, para escalonamento de sistemas Soft-RT, que considere, principalmente, a percepção dos usuários familiarizados ao modelo de atendimento preferencial da QoS relativa.

5.3.1 Descrição dos Cenários

Para os experimentos de simulação, foram especificadas duas classes de serviço: uma classe *A* mais exigente e outra classe *B* menos exigente. Vale lembrar que a exigência é relacionada ao ART especificado para cada classe e, no caso, quanto menor for essa média, maior será a exigência da classe. Esses cenários foram compostos por 20 usuários submetendo requisições ao sistema, sendo 10 usuários da classe *A* e 10 usuários da classe *B*. Sendo que todos os usuários, de uma mesma classe, possuem a mesma chance de fazer requisições *Web*, descrita por uma distribuição uniforme.

Para descrever tanto os intervalos de chegada quanto o tempo de execução das requisições, foi considerada uma carga de trabalho com distribuição exponencial, a mesma considerada em trabalhos anteriores, de forma a possibilitar a comparação dos resultados. Além disso, esse tipo de distribuição é amplamente utilizado na literatura para representar sistemas gerenciadores de filas (MacDougall, 1987). Essa distribuição tem como característica o cálculo das probabilidades para um certo tempo e espaço entre eventos sucessivos, sendo, portanto, comumente utilizada para a determinação dos tempos entre chegada das requisições. Nesse caso, os intervalos de chegada são descritos com média de 4 *u.t.*, enquanto que o tempo de execução apresenta 3 *u.t.*, equivalendo a um sistema com uma taxa de utilização de 75%, os mesmos valores considerados também em trabalhos anteriores.

O método proposto foi avaliado em diferentes cenários, os quais foram definidos, por meio da variação de alguns parâmetros, os mesmos considerados no desenvolvimento da

política EBS para análise se o método se apresenta ou não adequado nos cenários de ocorrência das inversões.

Um dos parâmetros que define os cenários é a porcentagem de variação de contrato (V), ou seja, a porcentagem de discrepância dos serviços, oferecidos por cada algoritmo em relação ao tempo de residência em um sistema de escalonamento convencional FIFO ($FIFO$). Vale ressaltar que utilizou-se, como base, ARTs que seriam oferecidos por um servidor convencional sem suporte à QoS, com o intuito de se definir contratos de serviços viáveis e não contratos com tempos impossíveis de serem garantidos com qualquer tipo de algoritmo de escalonamento.

Foram atribuídos a V os valores: 5%, 15% e 25%. Sendo que, quanto maior for a variação dos contratos em relação ao escalonamento FIFO, maior e menor será a dificuldade de atendimento dos usuários das classes A e B , respectivamente. As Equações 5.2 e 5.3 ilustram os cálculos dos contratos de A (C_A) e de B (C_B), respectivamente.

$$C_A = FIFO - (FIFO \cdot V) \quad (5.2)$$

$$C_B = FIFO + (FIFO \cdot V) \quad (5.3)$$

Outro parâmetro considerado é a proporção de requisições de cada classe no sistema. Por meio desse parâmetro especificou-se situações extremas, tais como um sistema sobrecarregado de requisições menos exigentes (por exemplo, 20%A-80%B) e um sistema sobrecarregado com requisições mais exigentes (por exemplo, 80%A-20%B e 90%A-10%B). Definiu-se também uma situação de igualdade na proporção de requisições de ambas as classes, ou seja, 50%A-50%B.

De forma a apresentar uma ampla amostragem dos dados, fixou-se como 100.000 o número de requisições submetidas em cada cenário simulado. Além disso, com o intuito de se obter melhor confiabilidade estatística, a simulação foi executada 30 vezes (Bhattacharyya & Johnson, 1977), utilizando-se as 30 diferentes sementes, i.e., replicações com fluxos de números aleatórios diferentes, disponibilizadas pelo SimpackJ. Sendo assim, os resultados obtidos foram analisados, segundo a média e o intervalo de confiança.

5.4 Resultados

Nessa Seção são analisados e discutidos os resultados obtidos pela simulação dos cenários apresentados na Seção anterior. A partir dos resultados obtidos, alguns pontos importantes devem ser salientados:

- verificação de quais fatores do sistema influenciam diretamente no comportamento do escalonador, e de que forma;

- análise das situações em que se consegue ganhos consideráveis de desempenho com o emprego do método proposto, determinando se o emprego desse método oferece alguma vantagem sobre a utilização da política EBS.
- inferir as taxas de satisfação dos usuários das diversas classes perante o uso do método proposto e da política de escalonamento EBS.

De forma a abranger essas questões os dados coletados foram organizados em três tipos de gráficos. O primeiro tipo tem por objetivo ilustrar os comportamentos da política EBS e do método proposto, mostrando para cada cenário qual o ART oferecido a um determinado usuário ao longo do tempo. Além de possibilitar a análise da eficácia do método em cumprir os contratos estabelecidos, este tipo de gráfico apresenta também um comparativo do desempenho entre o método proposto e a política EBS no que se refere à diferenciação dos serviços oferecidos aos usuários.

O segundo tipo de gráfico representa a variação da satisfação média dos usuários do sistema perante a aplicação da política EBS e do método proposto, sob a variação de diversos fatores do ambiente, como proporção de requisições entre as classes e valores de exigência contratual.

Por fim, o terceiro tipo de gráfico tem por objetivo mostrar a qualidade dos valores individuais de ARTs oferecidos, em média, para todos os usuários do sistema, apresentando a porcentagem média de ARTs de sistema que ficaram abaixo da contratado para cada usuário.

Para uma melhor organização, os resultados são apresentados nas Seções 5.4.1 e 5.4.2, descrevendo o comportamento do escalonador e a satisfação e variação do atendimento, respectivamente.

5.4.1 Análise do Comportamento do Escalonador

Nessa Seção, as Figuras 5.3 a 5.8 apresentam os experimentos de simulação que permitem um comparativo de comportamento entre a política EBS e o método proposto. Em seus gráficos são ilustrados os ARTs do sistema oferecidos ao longo de tempo para as classes de serviço mais e menos exigentes, classes A e B , respectivamente. O eixo das abscissas informa o término de atendimento de uma determinada requisição e o momento em que o ART efetivamente oferecido àquele usuário é atualizado. O eixo das ordenadas representam tais ARTs. Os valores dos contratos que definem as classes de serviço (A e B) são representados pelas retas horizontais C_A e C_B , respectivamente.

Foram selecionados alguns cenários principais de proporção de requisições e variação de contrato, dentre todos os estudados, de forma a analisar a influência destes parâmetros sobre a eficácia do método desenvolvido em respeitar os tempos de resposta estabelecidos nos contratos de cada classe, bem como a diferenciação entre elas.

5% Variação Contratual

Nos cenários apresentados pelas Figuras 5.3 (a) e 5.3 (b), há pouca proporção de requisições de classe *A*, apenas 20% e 50%, respectivamente, e a discrepância entre os valores contratuais é mínima, apenas 5% em relação a um escalonamento convencional. Conforme pode-se observar, pela Figura 5.3 (a), embora não ocorra o problema da inversão de prioridade, o método proposto começa a atuar nesses cenários, com o intuito de se diferenciar melhor os serviços oferecidos às classes. Sendo assim, a qualidade dos ARTs oferecidos a uma classe mais exigente utilizando-se o método proposto é um pouco melhor se comparada com os oferecidos pela política EBS. Com isso, os ARTs oferecidos a um usuário menos exigente com a utilização do método proposto mostrou-se um pouco acima dos ARTs oferecidos pela política EBS.

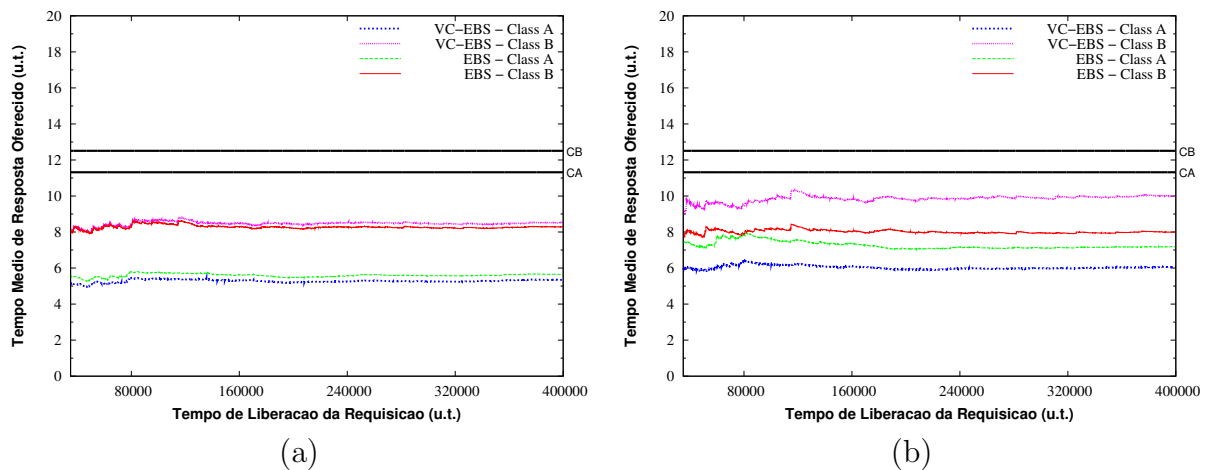


Figura 5.3: Comparativo entre a política EBS e o método proposto. Cenários com 5% de variação contratual e proporção de requisições: (a) 20%A e 80%B. (b) 50%A e 50%B.

Considerando o cenário apresentado pela Figura 5.3 (b), em que mantém-se a variação contratual e há uma igualdade na proporção de requisições (50%A-50%B), percebe-se uma aproximação dos ARTs oferecidos tanto para usuários de classe *A* quanto aos de classe *B*. Dessa forma, como o objetivo da utilização do contrato virtual é diferenciar melhor os serviços oferecidos às diferentes classes, nesse cenário, mesmo não ocorrendo a inversão de prioridade, o contrato virtual passa a atuar mais intensamente em comparação ao cenário anterior, conforme Figura 5.3 (a). A qualidade dos ARTs oferecidos a uma classe mais exigente, com o método proposto, é melhor se comparada com os oferecidos pela política EBS. Já os ARTs oferecidos a um usuário menos exigente mostrou-se acima dos ARTs oferecidos pela política EBS. Entretanto, esse aumento manteve-se abaixo da média contratada (C_B).

Por fim, para verificar o comportamento em um cenário totalmente saturado por requisições com baixa média contratada e submetido à mesma variação contratual descrita nos cenários anteriores executou-se os experimentos ilustrados pela Figura 5.4, em que o

sistema está sobrecarregado por requisições do tipo *A* em 80% e 90%, respectivamente.

Pode-se observar pelos gráficos da Figura 5.4 que mesmo com menor exigência contratual, um usuário do tipo *B* obtém melhores valores de ARTs do sistema, utilizando-se a política EBS, o que sinaliza uma inversão de prioridade em termos de QoS relativa. Isto ocorre pelo fato da diferença entre as exigências das classes ser muito pequena, o que dificulta a distinção do grau de urgência das requisições por parte do escalonador. Além disso, outro fator preponderante para os melhores ARTs oferecidos à requisições do tipo *B* é que o número destas no sistema é menor se comparado aos das de classe *A*.

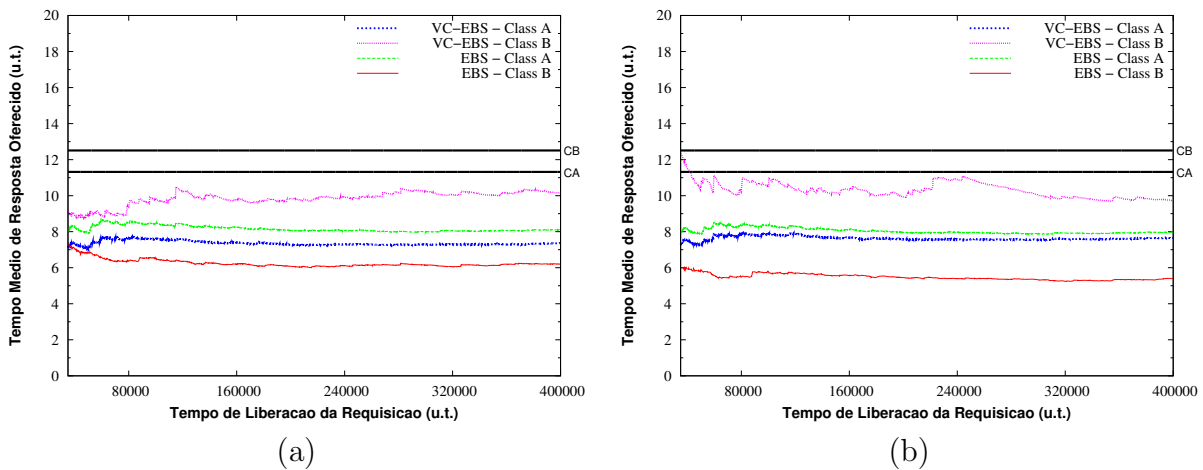


Figura 5.4: Comparativo entre a política EBS e o método proposto. Cenários com 5% de variação contratual e proporção de requisições: (a) 80%A e 20%B. (b) 90%A e 10%B.

No entanto, esta inversão de prioridade não influencia negativamente no cumprimento dos contratos. Os ARTs do sistema oferecidos para ambas as classes são melhores do que os exigidos, o que mostra o bom desempenho proporcionado pela política EBS em termos de garantia de QoS absoluta.

Já em termos de QoS relativa, pode-se observar pelos gráficos da Figura 5.4, a atuação do contrato virtual, revertendo o problema da ocorrência de inversão de prioridade. Os ARTs oferecidos a um usuário menos exigente ficou acima dos ARTs oferecidos pela EBS, enquanto que o oferecido a um usuário mais exigente teve uma considerável diminuição. É válido ressaltar que mesmo com uma proporção maior de requisições da classe *A*, requisições estas mais exigentes, o método proposto consegue garantir que os ARTs oferecidos sejam mantidos abaixo da média contratada, além de diferenciar melhor os serviços oferecidos às classes *A* e *B*.

15% Variação Contratual

A fim de avaliar a influência das exigências dos serviços disponibilizados sobre a qualidade do escalonamento, estabeleceu-se variações contratuais de 15% em relação a FIFO. Estabelecer uma variação contratual de 15% em relação ao tempo de residência oferecido

pela FIFO significa que o escalonador EBS pode trabalhar com a meta de garantir um ART do sistema, 15% melhor para requisições mais exigentes (classe *A*). Para que isto ocorra, o contrato da classe *B* precisa oferecer uma tolerância maior em seus valores de residência média no sistema, de forma a contrabalançar a demanda de desempenho.

Para verificar como este aumento de tolerância é percebido pelos usuários da classe *B*, ou seja, quanto que seus ARTs pioram para que usuários da classe *A* recebam melhor atendimento, simularam-se outros cenários propriamente ajustados, que podem ser visualizados a seguir. A partir das Figuras 5.5 e 5.6 mostra-se a capacidade do escalonador em garantir aumento da exigência e também a variação da qualidade do atendimento sob diferentes proporções entre classes.

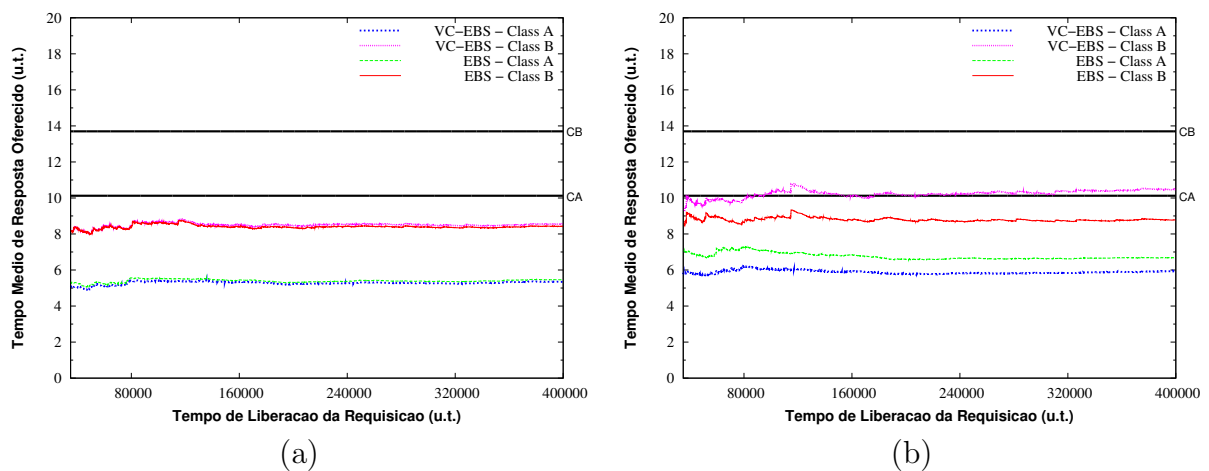


Figura 5.5: Comparativo entre a política EBS e o método proposto. Cenários com 15% de variação contratual e proporção de requisições: (a) 20%A e 80%B. (b) 50%A e 50%B.

Em cenários em que o número de requisições de alta prioridade é bastante inferior ao de baixa, como ilustra a Figura 5.5 (a), pode-se constatar que o contrato virtual não atua nesse cenário. Isso ocorre, pois, além da baixa proporção de requisições da classe *A* permitir que elas sejam melhor atendidas, a variação contratual nesse cenário é maior (15%), diferenciando melhor o atendimento entre as classes de serviços, não ocorrendo, portanto, a inversão de prioridade.

Mantendo-se a mesma variação contratual e incrementando o número de requisições mais exigentes no sistema, de forma a equilibrar um sistema com 50% de requisições mais exigentes e 50% de requisições menos exigentes, percebe-se uma aproximação dos ARTs oferecidos tanto para usuários de classe *A* quanto aos de classe *B*, como ilustrado na Figura 5.5 (b). Observa-se que com a utilização da política EBS, os ARTs oferecidos aos usuários de classe *A* aumentam e aos de classe *B* diminuem, ou seja, usuários menos exigentes começam a ter um atendimento preferencial, evidenciando a tendência em se ocorrer a inversão de prioridade a medida que a proporção de requisições mais exigentes aumenta.

Embora não ocorra a inversão de prioridade no cenário apresentado pela Figura 5.5

(b), o contrato virtual começa a atuar, diminuindo os ARTs oferecidos aos usuários mais exigentes e conseqüentemente levando a um aumento nos ARTs oferecidos aos usuários menos exigentes. Com a utilização do contrato virtual consegue-se, portanto, diferenciar melhor o atendimento entre as classes *A* e *B*.

Quando a proporção de requisições mais exigentes aumenta mais que a metade do total, como ilustam os gráficos da Figura 5.6, percebe-se um pequeno aumento dos ARTs oferecidos a esses usuários se comparado ao cenário anterior (Figura 5.5), em razão da maior exigência imposta ao escalonador. Valores de exigência maiores, quando potencializadas pelo aumento da proporção de requisições do tipo *A*, resultam em maiores cargas impostas ao sistema. No entanto, pode-se constatar também que os serviços oferecidos pela EBS e pelo método proposto, tanto para usuários da classe *A* quanto da classe *B* continuam com qualidade consideravelmente superior aos especificados pelos seus respectivos contratos.

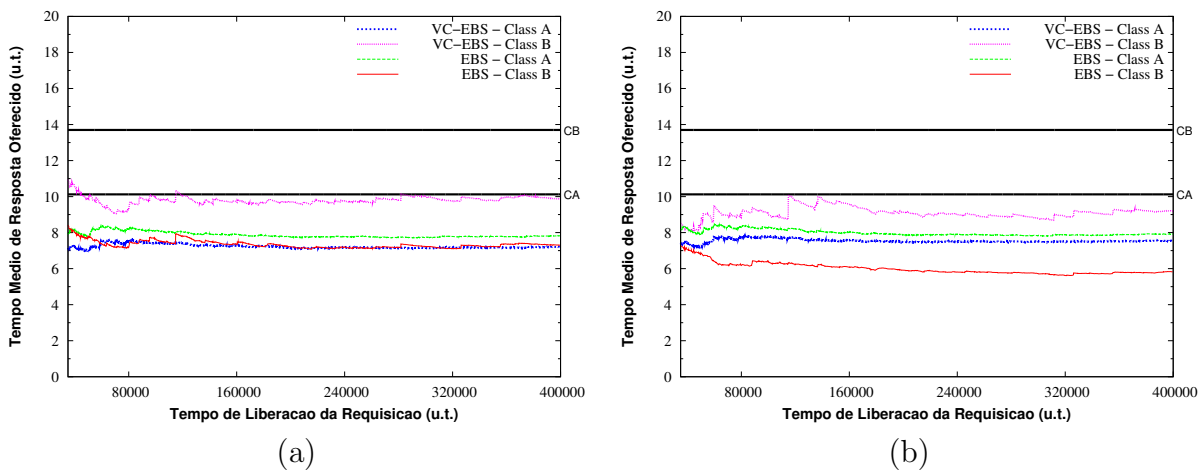


Figura 5.6: Comparativo entre a política EBS e o método proposto. Cenários com 15% de variação contratual e proporção de requisições: (a) 80%A e 20%B. (b) 90%A e 10%B.

Com a utilização do método proposto, além de se garantir a qualidade de serviço contratada, oferece-se também a diferenciação entre o atendimento oferecido às classes de serviço, ou seja, consegue-se reverter o problema da inversão de prioridade apresentada pela política EBS, conforme ilustrado em ambos os gráficos, Figura 5.6 (a) e Figura 5.6 (b).

25% Variação Contratual

Assim como em cenários anteriores (Figuras 5.3 (a) e 5.5 (a)), também se fixou a quantidade de requisições de alta prioridade a uma taxa bem inferior ao das de classe *B*, de forma a representar apenas 20% do total, porém com uma variação contratual de 25%, como ilustra a Figura 5.7 (a).

Esse cenário, se comparado com os anteriores do tipo, apresenta pouca alteração dos ARTs de ambas as classes, mesmo com considerável variação contratual. Uma das razões

que justificam isto é a baixa proporção de requisições de alta prioridade, que reduz o efeito da alta exigência sobre a carga total do sistema. Além desse, outro fator que contribui para a pequena alteração dos ARTs é a grande tolerância especificada para contratos de classe *B*.

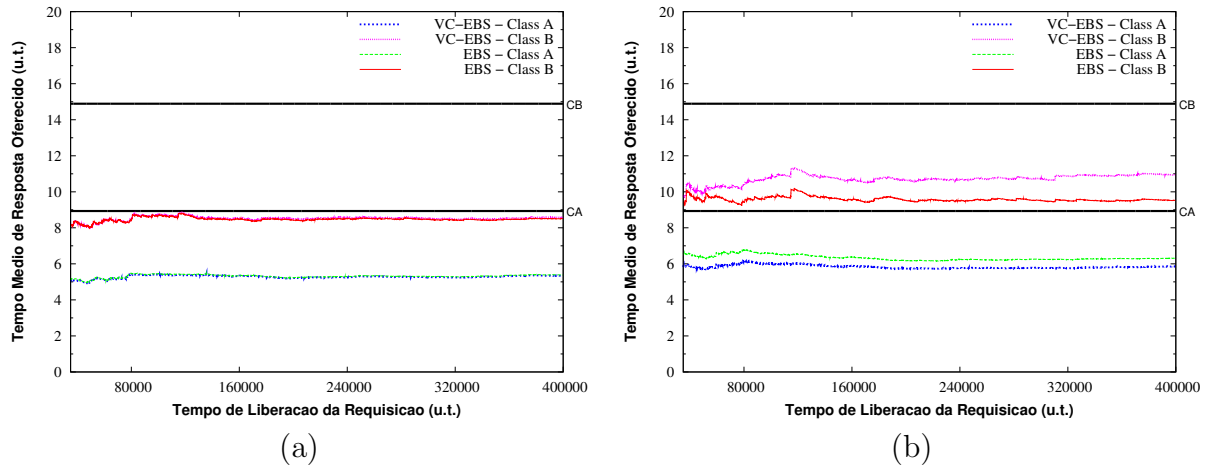


Figura 5.7: Comparativo entre a política EBS e o método proposto. Cenários com 25% de variação contratual e proporção de requisições: (a) 20%A e 80%B. (b) 50%A e 50%B.

Mantendo-se a mesma variação contratual (25%) e incrementando o número de requisições mais exigentes no sistema de 20% para 50% do total, percebe-se, utilizando a política EBS, um aumento dos ARTs do sistema oferecidos pelo escalonador tanto para usuários de classe *A* quanto aos de classe *B*, como ilustrado na Figura 5.7 (b).

Nesse cenário, Figura 5.7 (b), como nos cenários de mesma proporção apresentados anteriormente (Figuras 5.3 (b) e 5.5 (b)), embora não ocorra o problema da inversão de prioridade, o contrato virtual começa a atuar, porém de forma menos intensa. Essa intensidade é justificada pelo aumento do grau de diferenciação, devido à maior discrepância contratual, definida como forma de ser mais justa quanto ao nível de qualidade de serviço oferecido para cada classe.

Como discutido anteriormente, o aumento da proporção de requisições de alta prioridade tende a influenciar negativamente nos valores dos ARTs oferecidos pelos diversos serviços do sistema, pois a exigência imposta ao sistema tende à máxima. Quando o grau de exigência não é tão alto, como mostrado nos cenários com 5% de variação contratual, o peso imposto ao sistema é amortizado.

No entanto, como ilustrado na Figura 5.8, um aumento considerável da exigência contratual, em um cenário saturado de requisições de alta prioridade, tende a causar certo aumento dos ARTs das requisições de ambas as classes. As do tipo *A* são as mais afetadas, uma vez que o número de requisições de baixa prioridade é insuficiente para oferecer “brechas” de reordenação às requisições mais urgentes, que predominam na fila de escalonamento.

Mas mesmo em cenários saturados, a política EBS se mostra eficaz em garantir os

contratos para ambas as classes de serviço, além de diferenciar cada vez mais os serviços oferecidos à classes *A* e *B*, na medida em que se aumenta a variação contratual. Com isso, observa-se pela Figura 5.8, que embora trate-se de um caso de sobrecarga de requisições de alta prioridade, não ocorre a inversão de prioridade, sendo observada apenas em alguns momentos conforme ilustrado pelo gráfico da Figura 5.8 (b).

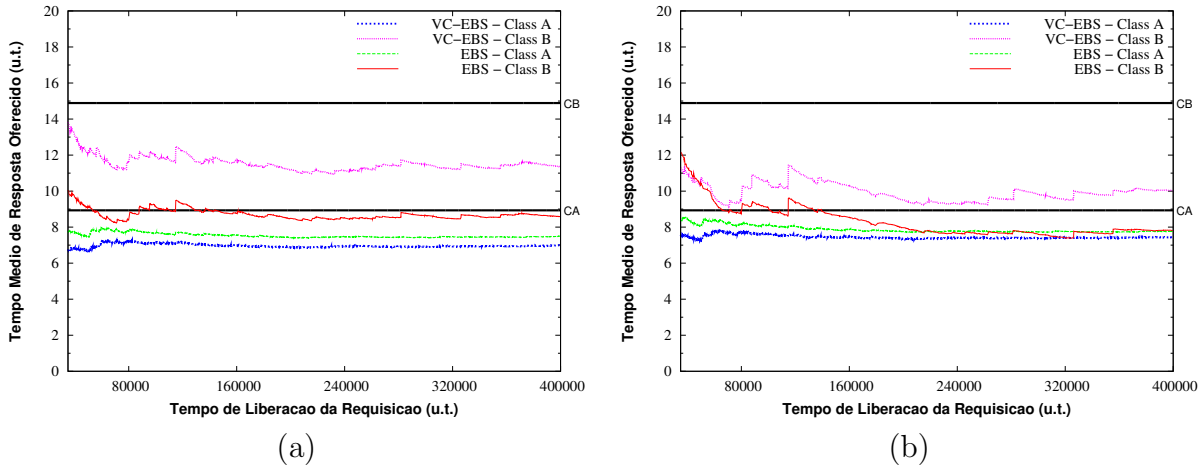


Figura 5.8: Comparativo entre a política EBS e o método proposto. Cenários com 25% de variação contratual e proporção de requisições: (a) 80%A e 20%B. (b) 90%A e 10%B.

Porém, em ambos os gráficos (Figura 5.8), nota-se que o contrato virtual atua priorizando as requisições mais exigentes e relaxando as restrições temporais dos usuários menos exigentes, de forma menos intensa se comparado aos cenários anteriores com menor variação contratual. Nota-se, portanto, que o contrato virtual atua cada vez menos, a medida que a variação contratual aumenta, dada a discrepância dos contratos das classes *A* e *B*.

5.4.2 Análise Quanto à Satisfação do Usuário

A análise dos resultados como apresentada na Seção 5.4.1, é importante pois fornece detalhes do comportamento do escalonador e da qualidade dos ARTs oferecidos. Porém, uma visão mais abrangente, com maior validade estatística também se faz necessária. Com isso, pode-se constatar se a eficiência e o desempenho observados nos gráficos anteriores são ou não casos isolados.

Além do ART, dados de satisfação e da variação de seus ARTs individuais em relação às médias contratadas foram coletados. Com relação à QoS absoluta, a satisfação de um determinado usuário mostra o percentual de vezes que o ART oferecido ficou de acordo com o estabelecido por cada contrato de serviço. Para medir a satisfação (S_i) dos usuários foi utilizada a Equação 5.4, que relaciona o número de vezes (N_i) que o ART das requisições ficou abaixo do contrato e o número total de requisições (R_i) submetidas pelo usuário i .

$$S_i = \frac{N_i}{R_i} \quad (5.4)$$

Enquanto que, em termos de QoS relativa, usuários mais exigentes estarão satisfeitos quando os ARTs oferecidos a eles forem menores que os ARTs oferecidos aos usuários menos exigentes. Nesse caso, a variável N_i da Equação 5.4, armazenará o número de vezes que os usuários mais exigentes estiveram satisfeitos.

Embora a satisfação, em média, possa se manter alta, os valores individuais de ARTs oferecidos podem variar muito. Para quantificar a variação da qualidade de um determinado atendimento, utilizou-se o número de ARTs oferecidos abaixo da média contratada. O percentual obtido reflete a estabilidade da qualidade do escalonamento oferecido.

A Figura 5.9 apresenta um comparativo da satisfação contratual dos usuários obtida pela utilização da política EBS e do método proposto (vc_{EBS}), considerando tanto QoS absoluta (Figura 5.9 (a)) quanto QoS relativa (Figura 5.9 (b)), em um cenário em que a proporção de requisições no sistema é composta por 20% da classe A e 80% da classe B.

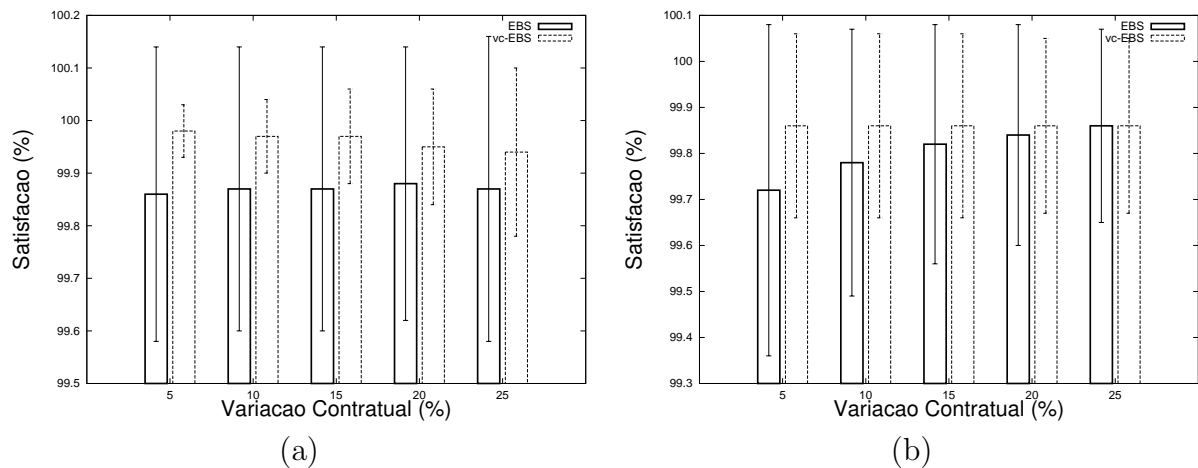


Figura 5.9: Gráfico representando a Satisfação Contratual dos usuários em termos de: (a) QoS Absoluta. (b) QoS Relativa em um cenário com 20% A e 80% B.

Quando a proporção de requisições prioritárias não é predominante no sistema, a política EBS mostra-se eficiente em garantir amplas taxas de diferenciação de serviço. A taxa de satisfação dos usuários da classe A é praticamente plena utilizando-se a política EBS e o método proposto e considerando tanto QoS absoluta como QoS relativa, verifica-se também alto grau de satisfação para todas as variações contratuais.

É possível também observar os desvios-padrão para as médias de satisfações apresentadas pela Figura 5.9. Diferentemente da política EBS, o método proposto consegue oferecer maior estabilidade da qualidade de seus serviços para todas as variações de exigências contratuais, e para ambos os gráficos. É importante ressaltar que os desvios apresentados são pequenos, embora não pareçam devido à escala dos gráficos.

Em casos em que o número de requisições prioritárias aumenta, mas não chega a ser

preponderante no sistema, apresentando uma participação de 50% das requisições totais, pode-se verificar como fica a satisfação dos usuários para diferentes níveis de exigência contratual (Figura 5.10). A satisfação proporcionada pela política EBS se mantém praticamente plena para todas as variações contratuais simuladas, considerando tanto QoS absoluta como relativa. Mesmo com o aumento da proporção de requisições prioritárias a satisfação oferecida pelo método proposto se manteve mais estável, se comparado com a política EBS.

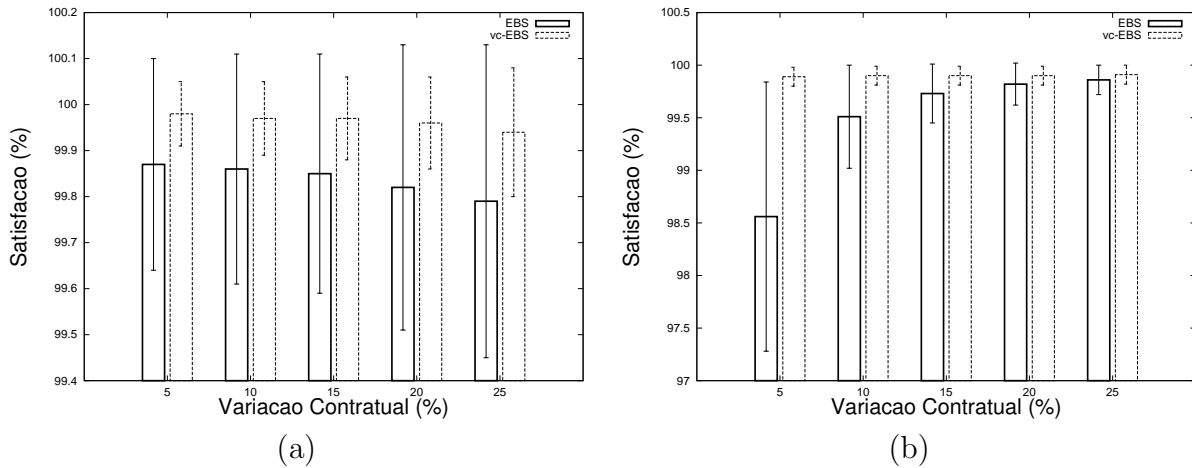


Figura 5.10: Gráfico representando a Satisfação Contratual dos usuários em termos de: (a) QoS Absoluta. (b) QoS Relativa em um cenário com 50% A e 50% B.

Quando o sistema fica saturado por requisições prioritárias, Figuras 5.11 e 5.12, a exigência imposta ao escalonador para garantir as requisições de QoS aumenta excessivamente. E a medida que a exigência contratual aumenta, pode-se observar que em termos de QoS absoluta (Figuras 5.11 (a) e 5.12 (a)), a satisfação dos usuários tende a diminuir, utilizando-se a política EBS. Com a utilização do método proposto têm-se uma maior estabilidade.

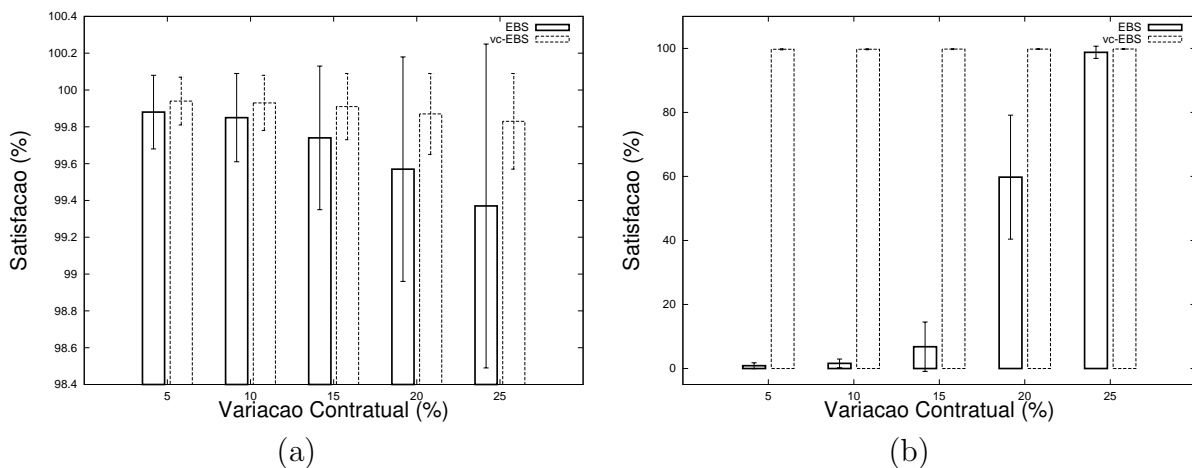


Figura 5.11: Gráfico representando a Satisfação Contratual dos usuários em termos de: (a) QoS Absoluta. (b) QoS Relativa em um cenário com 80% A e 20% B.

Já com relação à QoS relativa, constata-se pelas Figuras 5.11 (b) e 5.12 (b) que, com a utilização da política EBS, quanto menor for a discrepância contratual, menor é a satisfação dos usuários, devido a uma menor diferenciação dos serviços oferecidos a esses usuários. Variações de até 15% oferecem uma satisfação, aos usuários prioritários, inferior a 10%.

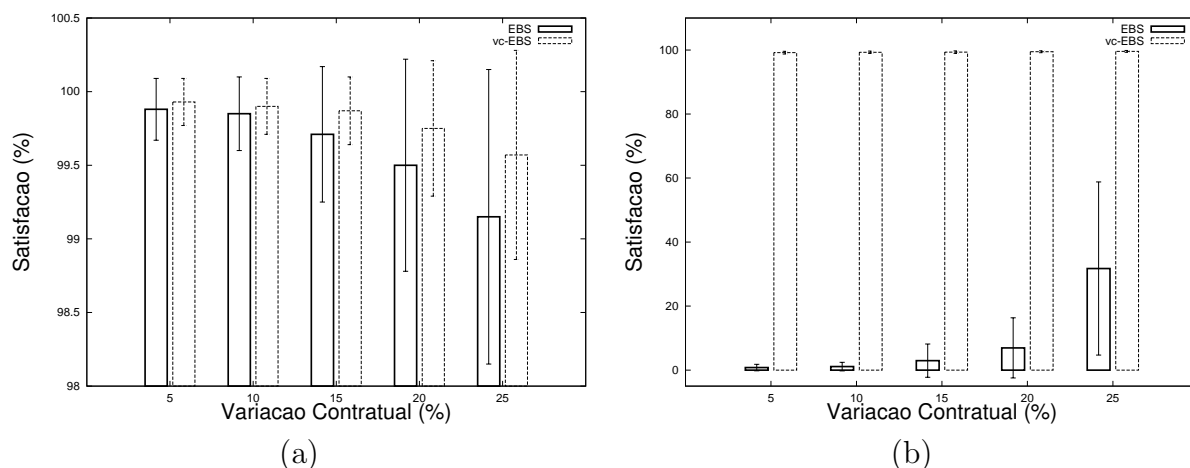


Figura 5.12: Gráfico representando a Satisfação Contratual dos usuários em termos de: (a) QoS Absoluta. (b) QoS Relativa em um cenário com 90% A e 10% B.

À medida que o grau de diferenciação cresce, a satisfação oferecida pela política EBS à classe *A* cresce, porém não atinge uma alta porcentagem de satisfação. Em um cenário sobrecarregado de 90% de requisições *A* e 10% de requisições *B*, conforme ilustrado pela Figura 5.12 (b), a satisfação dos usuários não atinge 60%, com uma variação de 25%. Com a utilização do método proposto, a satisfação mantém-se estável e praticamente plena para todas as variações contratuais simuladas. Os desvios-padrão apresentados (Figura 5.11 (b) e 5.12 (b)) se mantêm baixos e com pouca variação.

Essa maior estabilidade deve-se ao fato do método proposto se preocupar com o impacto da exigência de cada requisição a ser atendida sobre o sistema como um todo, diferenciando, portanto, o atendimento entre as classes de serviços e influenciando assim na satisfação dos usuários.

Pelas Figuras 5.13 e 5.14, pode-se verificar como varia a qualidade individual de atendimento das requisições para os cenários de 20%, 50%, 80% e 90% de requisições *A*, apresentados anteriormente. No escalonamento considerando tanto a política EBS como o método proposto, mesmo sob grandes variações contratuais das exigências, a qualidade do escalonamento se mantém em um bom nível. A qualidade individual dos tempos médios de resposta oferecidos é melhor que a contratada em mais de 80% dos casos.

É possível notar que para os mesmos valores de exigência contratual, a variação da qualidade dos valores individuais de tempo médio de resposta, em relação às diferentes proporções entre classes, é pequena. Quando a proporção de requisições da classe *A* é pequena (Figura 5.13 (a)), nota-se que o aumento de exigência contratual causa uma

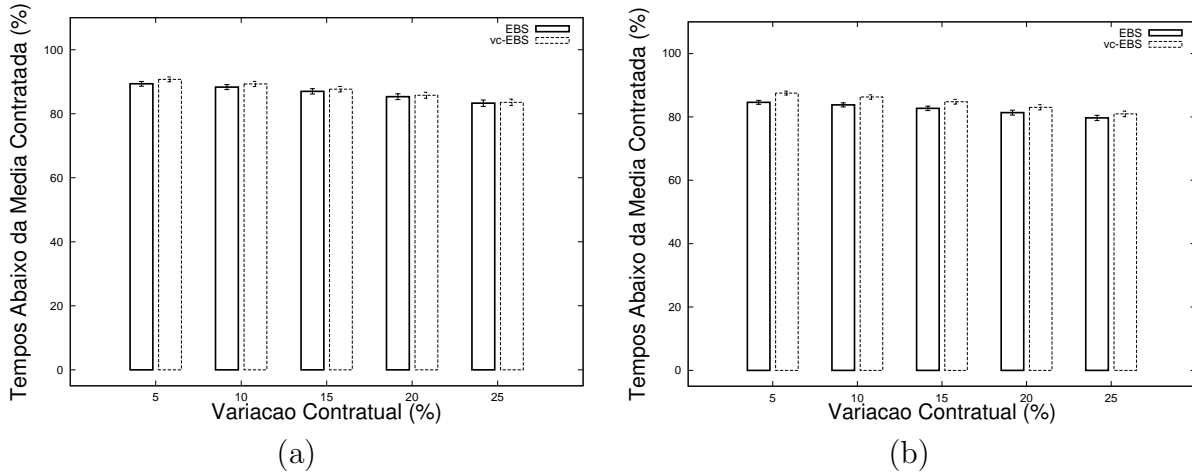


Figura 5.13: Gráfico representando os ARTs abaixo da média contratada. Cenários com: (a) 20% A e 80% B. (b) 50% A e 50% B.

pequena diminuição da taxa de tempos médios de resposta bem atendidos. À medida que o número de requisições prioritárias aumenta para 50% (Figura 5.13 (b)) do total, percebe-se o mesmo decréscimo de qualidade para taxas de classe A quando a exigência aumenta. Quando o sistema apresenta saturação de requisições prioritárias (Figuras 5.14 (a)) e 5.14 (b)), observa-se que as taxas de tempos médios de resposta caem ainda mais.

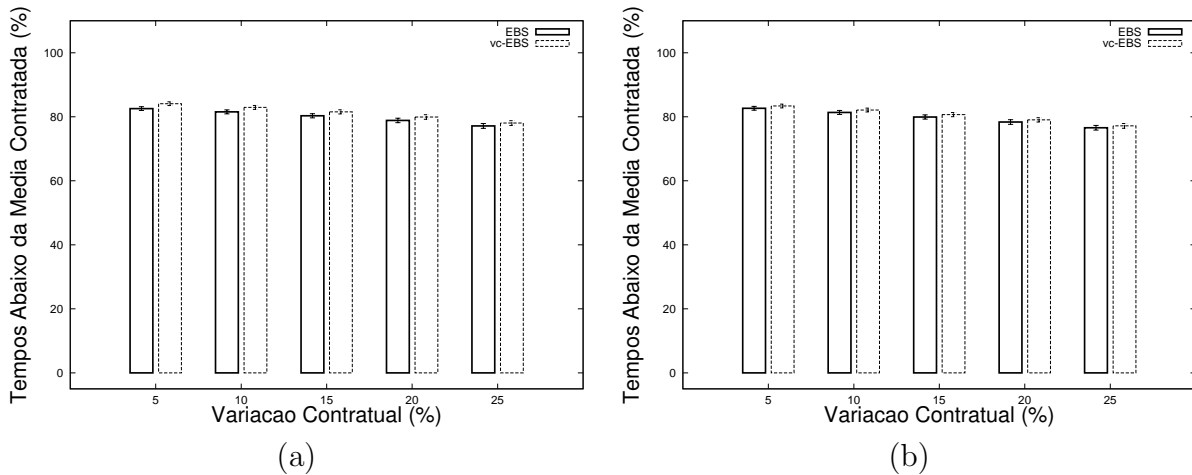


Figura 5.14: Gráfico representando os ARTs abaixo da média contratada. Cenários com: (a) 80% A e 20% B. (b) 90% A e 10% B.

5.5 Considerações Finais

No presente capítulo foi apresentado um novo método que emprega a técnica de *feedback scheduling* e considera contratos virtuais adaptativos de forma a relaxar as restrições dos usuários menos exigentes e priorizar as restrições dos usuários mais exigentes.

Para análise foram gerados três tipos de gráficos, ilustrando o comportamento do escalonador com a utilização da política EBS e do método proposto; a satisfação dos

usuários e a variação de seus ARTs individuais em relação às médias contratadas.

A partir dos experimentos realizados pôde-se notar claramente que o método proposto além de oferecer a garantia de desempenho aos usuários, atendendo-os de acordo com o que foi estabelecido em seus contratos e portanto garantindo a QoS absoluta, também diferencia o atendimento oferecido às diferentes classes de serviços, solucionando o problema da ocorrência de inversão de prioridade apresentada pela política EBS e garantindo, portanto, a QoS relativa, em um modelo composto por apenas duas classes de serviços. O Capítulo 6 apresenta a proposta de um novo modelo composto por n classes de serviços.

Proposta de um Novo Modelo Composto por n Classes de Serviços

6.1 Considerações Iniciais

Em vista à grande relevância atual da *Web*, como influente meio de comunicação e tecnologia fundamental para os sistemas de informação das mais avançadas empresas e organizações, faz-se necessário o aprimoramento na infraestrutura da *Web* para oferecer garantias de qualidade de serviço e atender assim as exigências das novas aplicações. Assim, em vista dessa motivação e do predomínio de trabalhos que abordam o tema no nível de rede, Casagrande et al. (2007) formulou e avaliou uma abordagem para oferecimento de suporte a QoS também no nível de aplicação (servidor *Web*), contribuindo, assim, para a implementação da qualidade fim-a-fim.

Diferentemente da maioria dos trabalhos atuais sobre o tema, a especificação se deu em termos absolutos, com a qual se pode oferecer garantias mais estritas de qualidade, atendendo assim às necessidades do número crescente de aplicações com requisitos responsivos tais como: comércio eletrônico, tele-medicina, sistemas multimídia, telefonia, ensino a distância e demais aplicações *online*. O foco da abordagem está em políticas de escalonamento que ofereçam garantias quantitativas de tempos de resposta especificadas em termos estocásticos. Tal garantia objetiva a provisão de valores médios de tempos de resposta de sistemas dentro de uma faixa de tolerância especificados por contratos de serviço firmados entre o usuário e o provedor *Web*.

No entanto, o problema estudado por Casagrande et al. (2007) considera um modelo de

apenas duas classes de serviços (ver Seção 3.3.2). Em termos de elaboração de modelos de negócios de provedores de serviços, seria conveniente considerar um sistema que abrangesse um número maior de classes. Neste capítulo é introduzida a proposta de um novo modelo composto por n classes de serviços.

6.2 Abordagem Proposta

A diferenciação de serviços é importante para acomodar expectativas e requisitos heterogêneos de usuários e aplicações, permitindo que planos de atendimento distintos sejam fixados para serviços na Internet (Blake et al., 1998). No nível de rede, por exemplo, agrupados em classes de serviço, fluxos de requisições/pacotes com necessidade de menores tempos de resposta e melhores taxas de entrega dos dados, como VoIP e videoconferência, podem ter seus requisitos operacionais garantidos, enquanto usuários com requisitos simples de conectividade, como transferência de dados hipertexto, recebem serviço de melhor esforço padrão (Xiao & Ni, 1999).

No nível de aplicação a utilização de mecanismos de diferenciação permite um maior controle, a QoS diferenciada poderia ser resultado de políticas de escalonamento, no caso favorecendo-se usuários pagantes em detrimento dos não pagantes (Eggert & Heidemann, 1999) ou baseadas em certos critérios para distinção de serviços oferecidos, como prover melhor qualidade aos usuários que arcam com os maiores custos (Vasiliou, 2000). Essas classes podem oferecer serviços simples, ou planos que podem ser classificados em *Gold*, *Silver* e *Bronze* (qualidade decrescente).

Para que essa diferenciação seja oferecida é preciso implantar mecanismos de controle capazes de gerenciar recursos e manter a qualidade de atendimento. No caso de QoS em nível de aplicação, abordagem deste trabalho, o emprego de tais mecanismos se faz por meio de algoritmos de escalonamento, que se utilizam de parâmetros e métricas que ofereçam informações de desempenho e qualidade de atendimento para tomar decisões, comparar resultados e tentar cumprir especificações de serviço. Esta seção tem como finalidade definir os parâmetros e as métricas utilizados para estruturar a abordagem proposta para n classes de serviços, visando comparar o comportamento da política EBS original para 2 classes de serviços e a nova abordagem proposta.

6.2.1 Descrição dos Cenários

Para os experimentos de simulação, foram especificadas n classes de serviços. Vale lembrar que a exigência é relacionada ao ART especificado para cada classe e, no caso, quanto menor for essa média, maior será a exigência da classe. Esses cenários foram compostos por $(n * 10)$ usuários submetendo requisições ao sistema, sendo 10 usuários em cada classe. Todos os usuários, de uma mesma classe, possuem a mesma chance de fazer requisições

Web, descrita por uma distribuição uniforme.

Dentro de cada classe, cada um dos seus 10 usuários exibe o mesmo modelo de geração de requisições, caracterizado pelo intervalo entre chegadas e o tempo de execução da requisição, ambos descritos pelas distribuições exponenciais com médias 4 u.t. e 3 u.t., respectivamente. Assim, o sistema continua com uma taxa de utilização de 75%.

Os cenários investigados nesse experimento consideram dois parâmetros: a proporção de requisições pertencentes a cada classe e a diferença entre o tempo médio de resposta contratado das classes de serviço.

Para o primeiro parâmetro, especificou-se duas situações extremas, uma com um sistema sobrecarregado de requisições de baixa exigência (10%A-45%B-45%C) e outra com um sistema sobrecarregado com requisições de alta exigência (90%A-5%B-5%C). Definiu-se também uma situação de igualdade no número de requisições de cada uma das classes (34%A-33%B-33%C).

Para o segundo parâmetro, definiu-se a porcentagem de variação de contrato (V) oferecidos tendo como referência o tempo médio de resposta do sistema com a simulação da política FIFO. Foram atribuídos a V , os valores: 5% e 15%. As Equações 6.1 a 6.3 descrevem os cálculos dos contratos de A (C_A), B (C_B) e C (C_C), respectivamente.

$$C_A = FIFO - (FIFO \cdot V) \quad (6.1)$$

$$C_B = FIFO \quad (6.2)$$

$$C_C = FIFO + (FIFO \cdot V) \quad (6.3)$$

Considerando quatro classes de serviços, os contratos das classes A , B , C e D , seriam, respectivamente, conforme Equações 6.4 a 6.7, e assim sucessivamente para um número maior de classes de serviços considerada.

$$C_A = FIFO - (FIFO \cdot (2 \cdot V)) \quad (6.4)$$

$$C_B = FIFO - (FIFO \cdot V) \quad (6.5)$$

$$C_C = FIFO + (FIFO \cdot V) \quad (6.6)$$

$$C_D = FIFO + (FIFO \cdot (2 \cdot V)) \quad (6.7)$$

6.2.2 Resultados

Nessa Seção são analisados e discutidos os resultados obtidos pela simulação dos cenários apresentados na Seção anterior. As Figuras 6.1 a 6.3 apresentam os experimentos de simulação que permitem um comparativo de comportamento entre a política EBS (modelo para 2 classes) e a abordagem proposta (modelo para n classes). Além disso, é realizado um comparativo entre o comportamento da abordagem proposta com a política sem suporte a QoS (FIFO). Em seus gráficos são ilustrados os ARTs do sistema oferecidos ao longo do tempo para cada uma das classes de serviços. O eixo das abscissas informa o término de atendimento de uma determinada requisição e o momento em que o ART efetivamente oferecido àquele usuário é atualizado. O eixo das ordenadas representam tais ARTs. Os valores dos contratos que definem as classes de serviços (A , B e C) são representados pelas retas horizontais C_A , C_B e C_C , respectivamente.

Foram selecionados alguns cenários principais de proporção de requisições e variação de contrato, dentre todos os estudados, de forma a analisar a influência destes parâmetros sobre a eficácia da abordagem proposta em respeitar os tempos de resposta estabelecidos nos contratos de cada uma das n classes. Além de verificar se o modelo de n classes proposto se comporta de acordo com o modelo de duas classes apresentado pela política EBS.

Nos cenários apresentados pela Figura 6.1, há pouca proporção de requisições da classe mais prioritária, apenas 10% e as discrepâncias entre os valores contratuais são de 5% e 25%, respectivamente, em relação a um escalonamento convencional FIFO. Como pode-se observar, a qualidade das médias oferecidas pela abordagem proposta ($nEBS$) é melhor se comparada com as oferecidas pela FIFO, da mesma forma que a apresentada pela política EBS original (modelo para duas classes de serviços). Os ganhos de desempenho e a qualidade de serviço oferecida em relação a FIFO são bem consideráveis. Outro fato que pode-se notar pela análise da Figura 6.1 é que as médias oferecidas pela FIFO não apresentam significativa diferença de qualidade entre as classes, pois o parâmetro de QoS não está relacionado ao mecanismo de priorização implementado por elas.

Em cenários em que a proporção de requisições mais exigentes no sistema é pequena, da mesma forma que a apresentada pela política EBS original (modelo para duas classes), nota-se a capacidade de garantia dos contratos de serviço por parte da abordagem proposta, pois as médias oferecidas para cada uma das classes ficaram em níveis significativamente melhores do que os contratados, garantindo portanto a QoS absoluta. Os baixos tempos de residência de sistema, devido ao menor impacto da exigência das classes sobre as médias de tempos de resposta, proporcionam menores tempos de resposta percebidos pelos usuários. Além disso, garante-se também a QoS relativa, o atendimento é oferecido de acordo com a classe em que o usuário se encontra, não ocorrendo portanto o problema de inversão de prioridade.

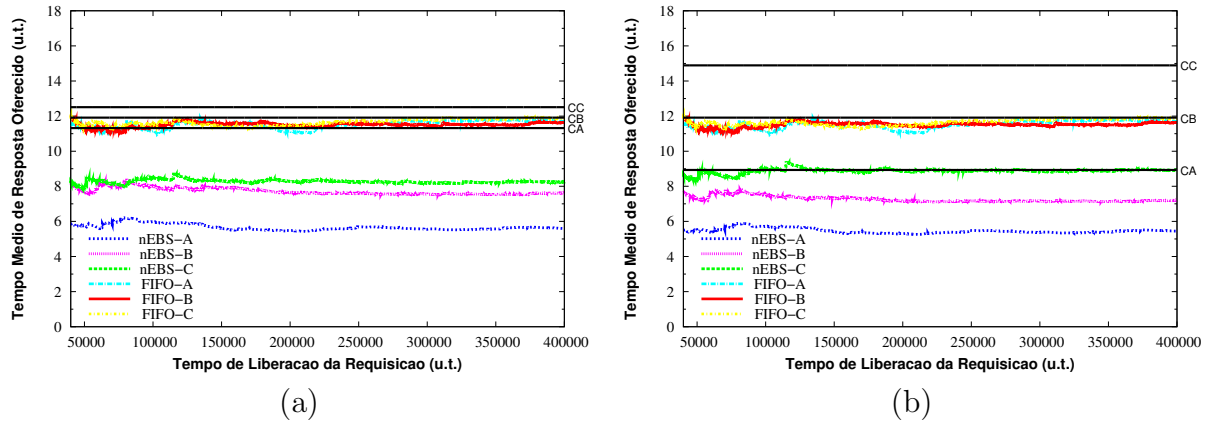


Figura 6.1: Representação do cenário de 10% de requisições A, 45% de requisições B e 45% de requisições C, apresentados pela abordagem proposta (*nEBS*) e pela política FIFO: (a) com 5% de variação de contrato. (b) com 25% de variação de contrato.

Com o aumento na discrepância entre os valores contratuais, Figura 6.1 (b), consegue-se diferenciar melhor o atendimento oferecido às n classes de serviços. As médias reais de tempo de resposta do sistema ficam melhores que as contratadas para ambos os tipos de serviço quando a abordagem proposta (*nEBS*) é empregada. Isto ocorre, pois, apesar da exigência da classe *A* ser relativamente alta, sua baixa proporção de requisições contrabalança a carga total imposta ao sistema. Estabelecendo uma variação contratual de 25% em relação a FIFO significa que o escalonador pode trabalhar com a meta de garantir um ART do sistema, 25% melhor para as requisições mais exigentes. Porém, para que isto ocorra, os contratos das classes menos exigentes precisam oferecer uma tolerância maior em seus valores de residência média no sistema, de forma a contrabalancear a demanda de desempenho.

Esta maior tolerância (menor carga) se reflete em um melhor escalonamento, pois a baixa probabilidade de existir requisições urgentes em fila diminui as chances de atrasos das demais. Isto possibilita que todos os contratos de serviço sejam devidamente cumpridos e com valores bem abaixo dos especificados. As requisições das classes menos exigentes apresentam uma qualidade de serviço melhor do que a especificada por contratos mais prioritários (como os da *A*).

A fim de verificar como este aumento de tolerância é percebido pelos usuários menos exigentes, ou seja, quanto que seus ARTs pioram para que usuários mais exigentes recebam melhor atendimento, simularam-se outros cenários, que podem ser visualizados a partir das Figuras 6.2 e 6.3. Por meio delas, mostra-se a capacidade do escalonador em garantir aumento da exigência e também a variação da qualidade do atendimento sob diferentes proporções entre classes.

Quando a proporção de requisições é a mesma para cada uma das classes e a discrepância entre os valores contratuais é pequena, como ilustra a Figura 6.2 (a), não há muita diferenciação entre os serviços, sendo assim, os ARTs oferecidos são semelhantes

(bastante próximos), utilizando-se tanto a abordagem proposta (nEBS) quanto a FIFO. Nesses cenários, em algumas situações verifica-se a ocorrência do problema de inversão de prioridade. Nesse caso, o contrato virtual pode ser adaptado de forma a atuar nesse novo modelo composto por n classes de serviços também.

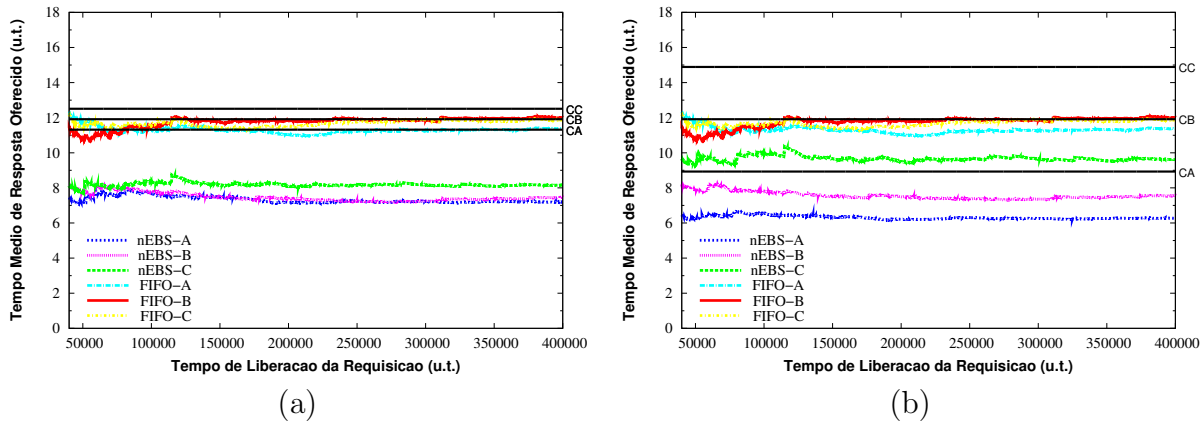


Figura 6.2: Representação do cenário de 34% de requisições A, 33% de requisições B e 33% de requisições C, apresentados pela abordagem proposta (*nEBS*) e pela política FIFO: (a) com 5% de variação de contrato. (b) com 25% de variação de contrato.

A medida que se aumenta a variação contratual (Figura 6.2 (b)), consegue-se diferenciar melhor os serviços oferecidos às n classes, priorizando as requisições mais exigentes, oferecendo, assim, melhores ARTs e consequentemente relaxando as restrições temporais dos usuários menos exigentes, aumentando os ARTs oferecidos a esses usuários. Nesse caso, não ocorre o problema de inversão de prioridade.

Nesse mesmo cenário (Figura 6.2), da mesma forma que o apresentado pela política EBS original (modelo para duas classes), percebe-se um aumento dos ARTs oferecidos se comparado ao cenário anterior (Figura 6.1), devido a maior exigência imposta ao escalonador e visto que a proporção de requisições mais exigentes aumenta de 10% para 34%. Valores de exigência maiores, quando potencializadas pelo aumento da proporção de requisições do tipo *A*, resultam em maiores cargas impostas ao sistema. No entanto, pode-se constatar também que os serviços oferecidos, pela abordagem proposta (nEBS) para os usuários de cada uma das n classes, continuam com qualidade consideravelmente superior aos especificados pelos seus respectivos contratos.

O aumento da proporção de requisições mais exigentes deixando um sistema sobrecarregado por esse tipo de requisição (Figura 6.3) e principalmente com uma variação contratual pequena, caracteriza um cenário de ocorrência do problema de inversão de prioridade. Nesses cenários, da mesma forma que o modelo para duas classes de serviços, a idéia dos contratos virtuais adaptativos pode ser aplicada nesse novo modelo composto por n classes.

Como ilustrado na Figura 6.3, o aumento na proporção de requisições mais exigentes tende a influenciar negativamente nos valores de média oferecidos pelos diversos serviços do

sistema, causando um certo aumento das médias de residência no sistema das requisições de cada uma das classes, pois a exigência imposta ao sistema tende à máxima. As mais exigentes são as mais afetadas, uma vez que o número de requisições menos exigentes é menor.

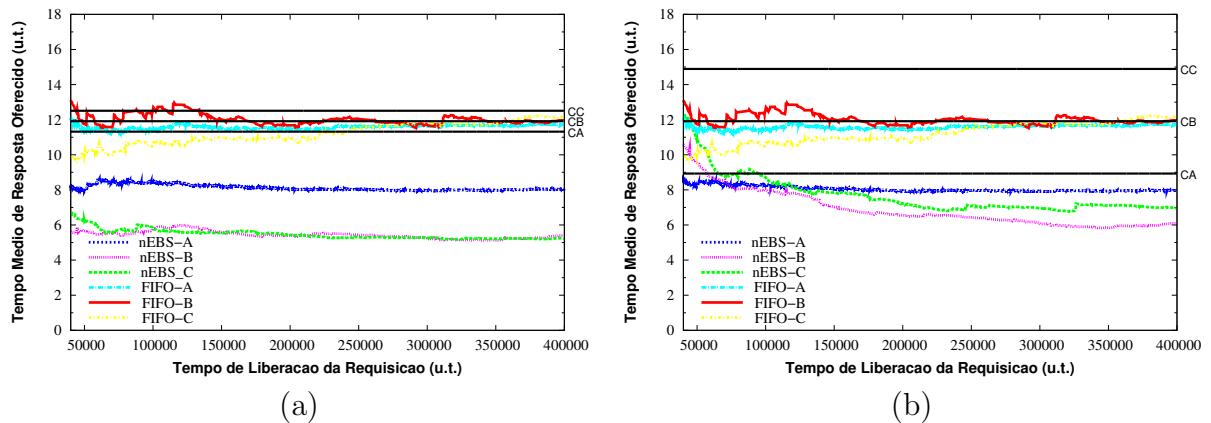


Figura 6.3: Representação do cenário de 90% de requisições A, 5% de requisições B e 5% de requisições C, apresentados pela abordagem proposta (*nEBS*) e pela política FIFO: (a) com 5% de variação de contrato. (b) com 25% de variação de contrato.

Mas mesmo em cenários saturados, a abordagem proposta, da mesma forma que a política EBS original (modelo para duas classes de serviços), se mostra eficaz em garantir as margens contratuais para cada uma das classes.

Tanto a política EBS quanto a abordagem (modelo para n classes) proposta, ao ponderar a urgência das requisições e o peso imposto ao sistema como um todo, conseguiu lidar melhor com a situação de sobrecarga, oferecendo médias de tempos de resposta melhores que as especificadas para cada uma das classes.

6.3 Considerações Finais

No presente capítulo foi apresentada uma nova abordagem de forma a estender a política EBS, cujo modelo de sistema é composto por apenas duas classes de serviços. A nova abordagem considera um modelo composto por n classes de serviços, a fim de representar um modelo de negócios de provedores de serviços mais prático para condições reais.

Os experimentos realizados demonstraram a eficácia da abordagem proposta em respeitar os tempos de resposta estabelecidos nos contratos de cada uma das n classes. Além de verificar que o modelo de n classes proposto se comporta de acordo com o previsto em estudos anteriores.

Trabalhos futuros poderão ser direcionados na verificação dos cenários em que observa-se a ocorrência da inversão de prioridade com a utilização do modelo proposto e posteriormente a aplicação do método de contratos virtuais adaptativos nesse modelo.

Conclusões

7.1 Considerações Gerais

A proposta desse trabalho de Mestrado foi contribuir para as áreas de qualidade de serviço e sistemas de tempo real, onde o ponto focal da presente dissertação esteve relacionado com a análise do problema de inversão de prioridade objetivando a melhor integração entre QoS relativa e absoluta. Nesse sentido, o principal resultado da pesquisa foi com relação a proposta de um novo método de escalonamento, apresentado no Capítulo 5, fundamentado na utilização de contratos virtuais adaptativos juntamente com a técnica de *feedback scheduling* com o intuito de melhorar o atendimento aos usuários, diferenciando os serviços oferecidos de acordo com suas respectivas classes, além de garantir desempenho de acordo com seus respectivos contratos. Tal método gerou uma publicação intitulada “*Escalonamento Realimentado para Diferenciação de Serviços e Garantia de Desempenho em Ambientes SOA com Requisitos Soft-RT*” no *X Simpósio em Sistemas Computacionais (WSCAD-SSC 2009)*.

A estratégia de escalonamento proposta permite um ajuste nos valores dos contratos dos usuários de forma a ponderar garantia de desempenho e maior grau de diferenciação de serviços, a medida que há um maior relaxamento das restrições temporais dos usuários menos exigentes (classe *B*) e uma maior priorização das restrições temporais dos usuários mais exigentes (classe *A*).

O desempenho dessa estratégia, em função tanto da sua capacidade em atender à determinadas classes de usuários como também em diferenciá-las, foi avaliado em diferentes situações de carga, principalmente naquelas em que há uma pequena variação contratual

e uma porcentagem maior de requisições mais exigentes, situações estas em que ocorrem as inversões.

Embora não ocorra o problema de inversão de prioridade, em cenários em que há pouca proporção de requisições mais exigentes e baixa discrepância entre os valores contratuais, o contrato virtual começa a atuar, com o intuito de se diferenciar melhor os serviços oferecidos às classes.

O aumento de requisições do tipo A reflete em uma pequena elevação do ART oferecido aos usuários dessa classe, devido à maior carga imposta ao sistema. Em contrapartida, a diminuição da proporção de requisições do tipo B influencia em um peso menor imposto por tal exigência de serviço, contribuindo para manter um bom ART oferecido aos seus usuários.

A medida que a proporção de requisições mais exigentes aumenta percebe-se uma aproximação dos ARTs oferecidos aos usuários das classes A e B , os ARTs oferecidos aos usuários da classe A aumentam e aos de classe B diminuem, ou seja, usuários menos exigentes começam a ter um atendimento preferencial, o que indica a tendência em ocorrer o problema da inversão de prioridade e portanto, o contrato virtual passa a atuar, cada vez mais, de forma intensa.

Em um cenário totalmente saturado por requisições com baixa média contratada, o qual sinaliza a ocorrência de inversão de prioridade em termos de QoS relativa, o método proposto consegue garantir que os ARTs oferecidos sejam mantidos abaixo da média contratada, além de diferenciar melhor os serviços oferecidos às classes A e B . A qualidade dos ARTs oferecidos a uma classe mais exigente, com o método proposto, é melhor se comparada com os oferecidos pela política EBS. Já os ARTs oferecidos a um usuário menos exigente mostrou-se acima dos ARTs oferecidos pela política EBS. No entanto, esse aumento manteve-se abaixo da média contratada (C_B).

A pequena discrepância entre as exigências é o fator principal que impede a garantia de boa diferenciação entre os serviços com a utilização da política EBS, deixando o nível de qualidade ser mais influenciado pela proporção de requisições de cada classe.

A medida que a variação contratual aumenta, ou seja, a diferença entre as exigências das classes passa a aumentar, facilita-se a distinção do grau de urgência das requisições por parte do escalonador e portanto, mesmo que haja uma proporção maior de requisições mais prioritárias no sistema, o contrato virtual passa a atuar de forma menos intensa.

Pela análise dos resultados apresentados no Capítulo 5, pode-se observar que o método proposto alcança com sucesso o objetivo inicialmente estabelecido: oferecer gerenciamento de QoS em termos relativos e absolutos para escalonamento no nível de aplicação. Conforme demonstrado, para a maioria dos cenários simulados os parâmetros contratuais de serviços de um sistema foram garantidos ao longo do tempo em patamares iguais ou abaixo dos contratados, mostrando eficiência e desempenho no cumprimento dos ARTs e na diferenciação dos serviços oferecidos às diferentes classes.

Além do tópico central da dissertação, o trabalho também derivou outra importante ramificação referente à proposta de uma nova abordagem de modelo de sistemas para a política EBS (modelo composto por apenas duas classes de serviços). A nova abordagem considera um modelo mais conveniente composto por n classes de serviços, representando um modelo de negócios de provedores de serviços mais prático para condições reais.

Por meio dos experimentos realizados, verificou-se a eficácia da abordagem proposta, apresentando o mesmo comportamento que a política EBS original (modelo de duas classes).

Além do projeto de pesquisa do mestrado, outro projeto de paralelização de algoritmos robóticos foi realizado juntamente com outros docentes do ICMC-USP. Esses projetos resultaram na submissão e publicação de dois artigos em eventos científicos internacionais e um artigo aceito para publicação em um periódico de alta seletividade.

7.2 Principais Contribuições

Resumidamente, as principais contribuições do presente trabalho, para o estado da arte nas áreas de sistemas computacionais de tempo real e de qualidade de serviço em nível de aplicação, ampliando os resultados já obtidos nessa linha de pesquisa, são descritas a seguir:

Primeiramente, foi realizada uma revisão crítica da bibliografia existente, a fim de contextualizar o presente trabalho e relatar os mais recentes avanços obtidos na área. Destaca-se especialmente o levantamento realizado sobre o fornecimento de QoS, em nível de aplicação nos servidores *Web*;

Devido à grande relevância atual e à demanda pelos diversos serviços da *Web*, como um influente meio de comunicação e tecnologia fundamental para os sistemas de informação das mais avançadas empresas e organizações, tornou-se essencial, além de oferecer garantias de qualidade de serviço e atender assim as especificações das novas aplicações, fornecer também diferenciação de serviços com diferentes níveis de qualidade para diferentes tipos de requisições, o qual foi realizado neste trabalho;

O método proposto neste trabalho permitiu uma abordagem inovadora de provisão de QoS em serviços *Web* ao relacionar o **gerenciamento da diferenciação de serviços** com a possibilidade de se conseguir **melhor qualidade de serviço**, oferecendo portanto **garantias tanto em termos de QoS relativa como absoluta** em uma mesma política de escalonamento.

Geralmente considera-se apenas características individuais nos mecanismos de provisão de qualidade de serviço, sem se preocupar com o impacto das decisões de escalonamento na qualidade dos demais serviços. No entanto, com a análise dos resultados obtidos pode-se perceber a importância do impacto da exigência imposta por uma classe sobre a qualidade de serviço oferecida para as demais, levando-se em consideração, principalmente,

a percepção dos usuários familiarizados ao modelo de atendimento preferencial da QoS relativa.

Em vista do predomínio de trabalhos que abordam o tema no nível de rede, o presente trabalho tem como contribuição a formulação e a avaliação de uma abordagem para o **oferecimento de suporte a QoS no nível de aplicação** (servidor *Web*), contribuindo, assim, para a implementação da qualidade fim-a-fim.

Este trabalho se propôs a investigar a elaboração de um novo método baseado em heurísticas para escalonamento de sistemas *Soft-RT*, de forma a caracterizar o atendimento do serviço, alcançando assim, uma contribuição também na **área de Sistemas não-determinísticos de Tempo Real**.

O novo método emprega a técnica de *feedback scheduling* e considera contratos virtuais adaptativos de forma a relaxar as restrições dos usuários menos exigentes e priorizar as restrições dos usuários mais exigentes, contribuindo, dessa forma, com a **solução da ocorrência de inversão de prioridade** apresentada pela política EBS.

Outra contribuição é que, sintetizando características de escalonamento de tempo real com baixos ARTs e de modelo realimentado, o método proposto oferece uma **abordagem multi-disciplinar** para provisão de novos níveis de confiabilidade às aplicações *Web*.

Além do método, o trabalho propõe também uma nova abordagem para o modelo apresentado pela política EBS (composto por duas classes). Propõe-se um novo modelo mais conveniente e mais prático para condições reais, considerando n classes de serviços.

Além da metodologia, o trabalho contribui ainda com um **conjunto de métricas e parâmetros** que possibilitam maior caracterização e estudo do impacto de provisão de QoS tanto em termos de QoS relativa quanto absoluta no escalonamento de requisições de um sistema.

Estas contribuições foram tema de artigo publicado em evento científico qualificado, o qual sintetiza os resultados e conclusões aqui relacionados.

- SAITO, P. T. M.; NOBILE, P. N.; MONACO, F. J. Escalonamento Realimentado para Diferenciação de Serviços e Garantia de Desempenho em Ambientes SOA com Requisitos Soft-RT. In: X Simpósio em Sistemas Computacionais (WSCAD-SSC), 2009, São Paulo, SP, Brasil. Anais do WSCAD-SSC. São Paulo, SP, Brasil: Sociedade Brasileira de Computação (SBC), p. 03-10, 2009.

Como trabalho adicional durante o período do mestrado, também foram publicados:

- SAITO, P. T. M.; SABATINE, R. J.; WOLF, D. F.; BRANCO, K. R. L. J. C. An Analysis of Parallel Approaches for a Mobile Robotic Self-localization Algorithm. International Journal of Future Generation Communication and Networking (IJFGCN), v. 2, p. 49-64, 2009.

- SAITO, P. T. M.; SABATINE, R. J.; WOLF, D. F.; BRANCO, K. R. L. J. C. Parallel Implementation of Mobile Robotic Self-localization. In: International Conference on Convergence and Hybrid Information Technology (ICHIT 2009), Daejeon, Korea. Proceedings of the 2009 ICHIT. New York, NY, USA: ACM International Conference Proceeding Series, 2009. v. 321. p. 390-396, 2009.
- SAITO, P. T. M.; MENDONCA, B. A.; SABATINE, R. J.; WOLF, D. F.; BRANCO, K. R. L. J. C. A Parallel Approach for Mobile Robotic Self-localization. In: International Conference on Computer Sciences and Convergence Information Technology (ICCIT), 2009, Seoul, Korea. Conference on Computer Sciences and Convergence Information Technology, p. 1-6, 2009.

7.3 Propostas para Trabalhos Futuros

A abordagem de escalonamento e o conjunto de métricas e parâmetros abordados neste projeto de mestrado consistem em um estudo inicial, podendo, portanto, serem ampliados. A seguir, são apresentadas algumas sugestões para trabalhos futuros de forma a dar continuidade ao trabalho apresentado nesta dissertação:

- Análise detalhada a fim de se descobrir nuances no comportamento da proposta do novo modelo composto por n classes de serviços, expondo, dessa forma, características ainda não descobertas;
- Utilização do método proposto baseado em contratos virtuais adaptativos para o novo modelo composto por n classes de serviços em situações de ocorrência de inversão de prioridade;
- Implementação do modelo de n classes e verificação da ocorrência de inversões de prioridade para possível implementação e aplicação do conceito de contrato virtual adaptativo;
- Investigação da política EBS, do novo modelo composto por n classes de serviços e o método proposto com uma taxa de utilização do sistema maior que 75%.
- Estudo e implementação do método e modelo propostos em um ambiente distribuído, verificando a viabilidade e os benefícios de estender o gerenciamento de QoS relativa e absoluta a um cluster de servidores *Web*, de forma que o emprego de uma quantidade maior de recurso ofereça a garantia de um bom atendimento (menores ARTs de sistema) a todos os usuários das diferentes classes de serviços, inclusive aos mais prioritários.

- Estudar a viabilidade de se implementar um módulo de controle de admissão para oferecer melhores níveis de confiabilidade dos requisitos de QoS das diversas classes do sistema.
- Análise da política EBS juntamente com o método proposto sobre contratos em janelas deslizante ou seções de operações, conforme pode ser observado em (Nery, 2008). Em sistemas reais, a qualidade de serviço apresentada em um grande período de atendimento pode representar uma satisfação contratual que não condiz com a experimentação do serviço. Entretanto, para o serviço operado com a EBS pode-se observar que mesmo com seções menores ainda é possível uma satisfação admissível para os usuários do sistema.

Referências Bibliográficas

- Abdelzaher, T. F., Shin, K. G., e Bhatti, N. (2002). Performance guarantees for web server end-systems: A control-theoretical approach. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):80–96.
- Almeida, J., Dabu, M., e Cao, P. (1998). Providing differentiated levels of service in web content hosting. In *First Workshop on Internet Server Performance - WISP (in conjunction with ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems)*, pp. 91–102.
- Andersson, B. e Tovar, E. (2006). Multiprocessor scheduling with few preemptions. In *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications - RTCSA*, pp. 322–334, Washington, DC, USA. IEEE Computer Society.
- Banks, J., Carson, J. S., e Nelson, B. L. (2000). *Discrete-Event System Simulation*. Prentice-Hall, Upper Saddle River, New Jersey 07458, 3 edition.
- Bhattacharyya, G. K. e Johnson, R. A. (1977). *Statistical concepts and methods*. Wiley, New York.
- Bhatti, N., Bouch, A., e Kuchinsky, A. (2000). Integrating user-perceived quality into web server design. In *Proceedings of the 9th international World Wide Web conference on Computer networks: the international journal of computer and telecommunications networking*, v. 33 of 1-6, pp. 1–16, Amsterdam, The Netherlands. North-Holland Publishing Co.
- Birtwhistle, G., Dahl, O., Myhrhaug, B., e Nygaard, K. (1979). *Simula Begin*. Chartwell-Bratt Ltd, 2 edition.
- Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., e Weiss, W. (1998). RFC 2475: An architecture for differentiated services. *Internet RFC, Internet Engineering Task Force - IETF*.
- Braden, R., Clark, D., e Shenker, S. (1994). RFC 1633: Integrated services in the internet architecture: an overview. *Internet RFC, Internet Engineering Task Force - IETF*.

- Cabral, M. I. C. e Souto, L. M. d. (2004). Especificação de componentes para a construção de simuladores de redes sem fio ad hoc padrão ieeee 802.11. *III Workshop de Dissertações da COPIN - WDCopin*, pp. 67–72.
- Calandrino, J. M., Baumberger, D., Li, T., Hahn, S., e Anderson, J. H. (2007). Soft real-time scheduling on performance asymmetric multicore platforms. *13th IEEE Real Time and Embedded Technology and Applications Symposium - RTAS*, 0:101–112.
- Casagrande, L. S. (2007). *Política de escalonamento de tempo real baseada em exigência para provisão de QoS absoluto em serviços Web*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil.
- Casagrande, L. S., de Mello, R. F., Bertagna, R., Andrade Filho, J. A., e Monaco, F. J. (2007). Exigency-based real-time scheduling policy to provide absolute QoS for web services. *19th International Symposium on Computer Architecture and High Performance Computing - SBAC-PAD*, 0:255–262.
- Chen, X. e Heidemann, J. (2003). Preferential treatment for short flows to reduce web latency. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 41(6):779–794.
- Cheng, A. M. K. (2002). *Real-Time Systems: Scheduling, Analysis, and Verification*. Wiley - Interscience, Hoboken, New Jersey.
- Cruz, G. M. e Lima, G. (2006). Simulador de escalonamento para sistemas de tempo real. *IV Workshop de Trabalhos de Iniciação Científica e de Graduação - WTICG-ERBASE*.
- Devi, U. e Anderson, J. (2006a). Flexible tardiness bounds for sporadic real-time task systems on multiprocessors. *20th International Parallel and Distributed Processing Symposium - IPDPS*, 0:1–10.
- Devi, U. e Anderson, J. (2006b). *Soft real-time scheduling on multiprocessors*. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
- Eggert, L. e Heidemann, J. (1999). Application-level differentiated services for web servers. *World Wide Web*, 2(3):133–142.
- Estrella, J. C., Teixeira, M. M., e Santana, M. J. (2006). Negotiation mechanisms on application level: a new approach to improve quality of service in web servers. *The 4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and 2nd International Workshop on Collaborative Computing, Integration, and Assurance. SEUS/WCCIA*, 0:255–260.
- Farines, J. M., Fraga, J. S., e Oliveira, R. S. (2000). *Sistemas de Tempo Real*, v. 1. Escola de Computação da Sociedade Brasileira de Computação, São Paulo, SP, 1 edition.
- Fishwick, P. A. (1992). Simpack: getting started with simulation programming in C and C++. In *Proceedings of the 24th conference on Winter simulation - WSC*, pp. 154–162, New York, NY, USA. ACM.
- Freitas, P. J. F. (2008). *Introdução à Modelagem e Simulação de Sistemas com Aplicações Arena*. Visual Books, 2 edition.

- Gomes, E. A. (2009). Sistemas operacionais: Escalonamento. notas de aula. pp. 1–8, Salvador, BA, Brasil. Universidade Federal da Bahia (UFBA). Disponível em: http://www.edeyson.com.br/Arquivos/SO/Aulas/SO_06_Escalonamento.pdf. Acesso em: janeiro de 2010.
- Henriksson, D., Lu, Y., e Abdelzaher, T. (2004). Improved prediction for web server delay control. *Proceedings of 16th Euromicro Conference on Real-Time Systems - ECRTS*, 0:61–68.
- Hlavicka, J. e Racek, S. (2002). C-Sim - the C language enhancement for discrete-time simulations. *Proceedings of the International Conference on Dependable Systems and Networks - DSN*, 0:539.
- ISO/IEC (1998). Information technology - quality of service: Framework. Technical report, ISO/IEC International Organization for Standardization/International Electrotechnical Commission 13236.
- Jain, R. (1991). *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling*. New York, NY, USA, Wiley.
- Jeffay, K., Stanat, D. F., e Martel, C. U. (1991). On non-preemptive scheduling of periodic and sporadic tasks. In *Proceedings of the 12th Real-Time Systems Symposium - RTSS*, pp. 129–139, San Antonio, Texas. IEEE Computer Society Press.
- Kang, K.-D., Son, S. H., e Stankovic, J. A. (2003). Differentiated real-time data services for e-commerce applications. *Electronic Commerce Research, Special Issue on Business Process Integration and E-Commerce Infrastructure*, 3(1-2):113–142.
- Kanodia, V. e Knightly, E. W. (2003). Ensuring latency targets in multiclass web servers. *IEEE Transactions on Parallel and Distributed Systems*, 14(1):84–93.
- Kato, S. e Yamasaki, N. (2008). Portioned EDF-based scheduling on multiprocessors. In *Proceedings of the 7th ACM International Conference on Embedded Software - EM-SOFT*, pp. 139–148, New York, NY, USA.
- Kermia, O. e Sorel, Y. (2008). Schedulability analysis for non-preemptive tasks under strict periodicity constraints. *International Workshop on Real-Time Computing Systems and Applications - RTCSA*, 0:25–32.
- Kumar, K. H. e Majhi, S. (2004). Queuing theory based open loop control of web server. In *Proceedings of the 2004 American Control Conference*, 3:2314–2315.
- Laplante, P. A. (2004). *Real-Time Systems Design and Analysis*. Wiley-IEEE Press, Hoboken, New Jersey, 3 edition.
- Law, A. M. e Kelton, W. (2006). *Simulation Modeling and Analysis*. McGraw-Hill Publishing Co., USA, 4 edition.
- Lazowska, E. D., Zahorjan, J., Graham, G. S., e Sevcik, K. C. (1984). *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*. Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1 edition.

- Lee, S. C. M., Lui, J. C. S., Member, S., e Yau, D. K. Y. (2003). A proportional-delay diffserv-enabled web server: Admission control and dynamic adaptation. *IEEE Transactions on Parallel and Distributed Systems*, 15(5):385–400.
- Leite, J. C. (2007). Eventos, tarefas, tempos e prazos. notas de aula. pp. 1–8, Natal, RN, Brasil. DIMAp - Universidade Federal do Rio Grande do Norte (UFRN). Disponível em: <http://www.ici.unifei.edu.br/ramos/download/STR/4.EventosTarefasTempoPrazos.pdf>. Acesso em: janeiro de 2010.
- Leontyev, H. e Anderson, J. H. (2007). Tardiness bounds for FIFO scheduling on multiprocessors. In *Proceedings of the 19th Euromicro Conference on Real-Time Systems - ECRTS*, pp. 71–81, Washington, DC, USA. IEEE Computer Society.
- Lin, T. H. e Tarng, W. (1991). Scheduling periodic and aperiodic tasks in hard real-time computing systems. *SIGMETRICS Performance Evaluation Review*, 19(1):31–38.
- Liu, C. L. e Layland, J. W. (2002). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Readings in hardware/software co-design*, pp. 179–194.
- Liu, J. W. S. (2000). *Real-time Systems*. Prentice Hall, Upper Saddle River, New Jersey 07458, 1 edition.
- Lu, C., Abdelzaher, T. F., Stankovic, J. A., e Son, S. H. (2001). A feedback control approach for guaranteeing relative delays in web servers. In *Proceedings of the 7th IEEE Real-Time and Embedded Technology and Applications Symposium - RTAS*, v. 0, pp. 51–63, Washington, DC, USA. IEEE Computer Society.
- MacDougall, M. H. (1987). *Simulating computer systems: techniques and tools*. MIT Press Series in Computer Systems, Cambridge, MA, USA.
- MacDougall, M. H. e McAlpine, J. S. (1973). Computer system simulation with ASPOL. In *Proceedings of the 1st Symposium on Simulation of Computer Systems - ANSS*, pp. 92–103, Piscataway, NJ, USA. IEEE Press.
- Magalhães, M. F. e Cardozo, E. (1999). Qualidade de serviço na internet. Technical report, Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial - UNICAMP, Campinas, SP.
- Martin, P. (1997). The development of an object-oriented, discrete-event simulation language using java. *4th Asia-Pacific Software Engineering and International Computer Science Conference - APSEC/ICSC*, 0:123.
- Mathes, M., Heinzl, S., e Freisleben, B. (2008). WS-temporal policy: A WS-policy extension for describing service properties with time constraints. *32nd Annual IEEE International Computer Software and Applications Conference - COMPSAC*, 0:1180–1186.
- Messias, V. R. (2007). *Servidor web distribuído com diferenciação de serviços - implementação e avaliação de um protótipo*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil.

- Messias, V. R., Estrella, J. C., Santana, M. J., Santana, R. H. C., Bruschi, S. M., e Teixeira, M. M. (2007). Implementação de um protótipo de servidor web distribuído com diferenciação de serviços. In *XIII Simpósio Brasileiro de Sistemas Multimídia e Web - WebMedia*, pp. 103–110.
- Monaco, F. J., Nery, M., e Peixoto, M. M. L. (2009). An orthogonal real-time scheduling architecture for responsiveness QoS requirements in SOA environments. In *Proceedings of the 2009 ACM symposium on Applied computing*, pp. 1–6, New York, NY, USA. ACM. to be published.
- Monaco, F. J. e Nobile, P. N. (2009). Feedback-based adaptive resource control in QoS-aware SOA systems with soft real-time requirements. In *III International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA)*, pp. 1–6.
- Nery, M. (2008). *Uma Política de Escalonamento de Tempo-Real para Garantias de QoS na Web Baseada em Parâmetros Média e Dispersão do Tempo de Resposta*. Qualificação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil.
- Nichols, K., Blake, S., Baker, F., e Black, D. (1999). RFC 2474: Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers. *Internet RFC, Internet Engineering Task Force - IETF*.
- Nissanke, N. (1997). *Realtime Systems*. Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1 edition.
- Pan, W., Mu, D., Wu, H., e Yao, L. (2008). Feedback control-based QoS guarantees in web application servers. In *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications - HPCC*, pp. 328–334, Los Alamitos, CA, USA. IEEE Computer Society.
- Peixoto, M. L. M. (2008). *Políticas de escalonamento de tempo-real para garantia de QoS absoluta em array de servidores web heterogêneos*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil.
- Peixoto, M. L. M., Tott, R., Nery, M., e Monaco, F. J. (2008). Arquitetura de escalonamento ortogonal de tempo-real para garantias de QoS em servidores web. In *Workshop em Desempenho de Sistemas Computacionais e de Computação - WPerformance*, pp. 18–37. Anais do XXVIII Congresso da SBC.
- Posse, E. e Vangheluwe, H. (2007). Kiltera: A simulation language for timed, dynamic structure systems. *40th Annual Simulation Symposium - ANSS*, 0:293–300.
- Pradhan, P., Tewari, R., Sahu, S., Ch, A., e Shenoy, P. (2002). An observation-based approach towards self-managing web servers. In *Proceedings of the 10th International Workshop on Quality of Service - IWQoS*, pp. 13–22, Miami Beach, Florida, USA.
- Prado, D. S. (2004). *Teoria das Filas e da Simulação*. Nova Lima, MG, Brasil, 2 edition.

- Schwetman, H. (1986). CSIM: A C-based, process-oriented simulation language. In *Proceedings of the 18th conference on Winter simulation - WSC*, pp. 387–396, New York, NY, USA. ACM.
- Sha, L., Abdelzaher, T., Arzén, K. E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., e Mok, A. K. (2004). Real time scheduling theory: A historical perspective. *Real-Time Systems*, 28(2-3):101–155.
- Shannon, R. E. (1998). Introduction to the art and science of simulation. In *Proceedings of the 30th Conference on Winter simulation - WSC*, pp. 7–14, Los Alamitos, CA, USA. IEEE Computer Society Press.
- Shi-jun, Z., Xin, Y., Shao-hua, Y., e Ben-xiong, H. (2008). A DMR fair algorithm for realtime scheduler. *2nd International Conference on Future Generation Communication and Networking - FGCN*, 1:362–366.
- Silva, L., Pereira, A., e Jr Meira, W. (2006). Reactivity-based scheduling approaches for internet services. *4th Latin American Web Congress - LA-WEB*, pp. 47–58.
- Soares, L. F. G. (1992). *Modelagem e Simulação Discreta de Sistemas*. Editora Campus Ltda, 1 edition.
- Stankovic, J. A., Spuri, M., Natale, M. D., e Buttazzo, G. C. (1995). Implications of classical scheduling results for real-time systems. *Computer*, 28(6):16–25.
- Strnadt, C. (2002). At your service: QoS for the internet. *IEEE Multimedia - Media Reviews*, 9(1):93–95.
- Tavares, E., Silva, B., Maciel, P., e Dallegrave, P. (2008). Software synthesis for hard real-time embedded systems with energy constraints. In *Proceedings of the 2008 20th International Symposium on Computer Architecture and High Performance Computing - SBAC-PAD*, pp. 115–122, Washington, DC, USA. IEEE Computer Society.
- Teixeira, M. M., Santana, M. J., e Santana, R. H. C. (2004). Using adaptive priority controls for service differentiation in QoS-enabled web servers. In *International Conference on Computational Science - ICCS*, v. 3036 of *Lecture Notes on Computer Science*, pp. 537–540, Cracóvia, Polónia. Springer.
- Teixeira, M. M., Santana, M. J., e Santana, R. H. C. (2005). Servidor web com diferenciação de serviços: Fornecendo qos para os serviços da internet. In *XXIII Simpósio Brasileiro de Redes de Computadores (SBRC)*, pp. 1–14, Fortaleza, CE.
- Tott, R. F. (2008). *Extensões na política EBS - Controle de admissão e redução da ordem de complexidade temporal*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil.
- Traldi, O. A., Barbato, A. K., e Santana, R. H. C. (2006). Service differentiating algorithms for QoS-enabled web servers. In *Proceedings of the 12th Brazilian Symposium on Multimedia and the Web - WebMedia*, pp. 263–272, New York, NY, USA. ACM.
- Vasiliou, N. (2000). Overview of internet QoS and web server QoS, reading course report. Department of Computer Science, The University of Western Ontario, London, Ontario, Canada.

- Wei, J. e Xu, C. (2005). A self-tuning fuzzy control approach for end-to-end QoS guarantees in web servers. In *Proceedings of the 13th International Workshop Quality of Service - IWQoS*, v. 3552 of *Lecture Notes in Computer Science*, pp. 123–135, Passau, Germany. Springer.
- Wei, Y., Ren, F., Lin, C., e Voigt, T. (2005). Fuzzy control for guaranteeing absolute delays in web servers. In *Proceedings of the 2nd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks - QSHINE*, pp. 48–51, Washington, DC, USA. IEEE Computer Society.
- Xiangbin, Z. (2008). Support QoS in open real-time systems. *International Conference on Computer Science and Software Engineering - CSSE*, 2:190–193.
- Xiao, X. e Ni, L., M. (1999). Internet QoS: A big picture. *IEEE Network*, 13(2):8–18.
- Ye, N., Gel, E. S., Li, X., Farley, T., e Lai, Y.-C. (2005). Web server QoS models: applying scheduling rules from production planning. *Computer and Operations Research*, 32(5):1147–1164.
- Ye, N., Li, X., Farley, T., e Xu, X. (2007). Job scheduling methods for reducing waiting time variance. *Computers and Operations Research*, 34(10):3069–3083.
- Zhang, Y., Kreckler, D. K., Gill, C., Lu, C., e Thaker, G. H. (2008). Practical schedulability analysis for generalized sporadic tasks in distributed real-time systems. In *Proceedings of the 20th Euromicro Conference on Real-Time Systems - ECRTS*, v. 0, pp. 223–232. IEEE Computer Society.
- Zhao, W., Olshefski, D., e Schulzrinne, H. (2000). Internet quality of service: an overview. Technical report, Technical Report CUCS-003-00, Columbia University, Computer Science Department.

Este documento foi preparado com o formatador de textos L^AT_EX .