# Reconhecimento de Sessões em um Modelo para Servidor Web com Diferenciação de Serviços

**Hima C. B. Mourão, Marcos J. Santana, Regina H. C. Santana, Alessandra K. Barbato**

Departamento de Sistemas e Computação – Universidade de São Paulo (USP)
Caixa Postal 668 1356-970 – São Carlos, SP – Brasil

{hcarla, mjs, rcs, akb}@icmc.usp.br

***Abstract.*** *This article presents simulation results of a new sessions admission control (CAS) incorporated in a web server model with differentiation of services (SWDS) to provide guarantees of sessions finalizations as of sites e-business. The new mechanisms for accepting new sessions, Hard-Threshold and Based in Model Session (BSM), had been developed for the CAS. Mechanism for the requests admission control (CAR) was developed to prevent overloads to the server model, beyond negotiation politics that they had supplied better and differentiated services to the sessions longest. The gotten results of simulation prove the functionality of the controls.*

***Resumo.*** *Este artigo apresenta resultados obtidos de simulações de um novo controle de admissão de sessões (CAS) incorporado a um modelo de servidor web com diferenciação de serviços (SWDS), para fornecer garantias de finalização de sessões como as de sites e-business. Foram desenvolvidos mecanismos de aceitação de novas sessões, Hard-Threshold e o Baseado em Sessão Modelo (BSM) ao CAS, e para o controle de admissão de requisições CAR novos mecanismos para prevenir sobrecargas ao modelo de servidor, além de políticas de negociação que forneceram melhores e diferenciados serviços às sessões mais longas, como os resultados obtidos comprovam.*

## 1. Introdução

As atuais aplicações encontradas na Web, como serviços bancários e *e-commerce*, demandam grandes esforços dos servidores web, justificando o intenso tráfego encontrado atualmente na Internet. Estas aplicações exigem ainda melhores condições de atendimento aos seus clientes, no entanto o modelo de melhor esforço (*best-effort*) adotado pela Internet não foi projetado para o uso observado atualmente, onde nenhuma garantia de serviço é oferecida às aplicações. Neste contexto, muitas pesquisas foram realizadas quanto ao fornecimento de qualidade de serviços (QoS) relativa a rede [Andreolini *et al.* 2004] [Blake *et al.* 1998]. Porém, a QoS é completamente obtida somente quando todos elementos da rede fornecem qualidade de serviço, inclusive os servidores web, considerados como elementos finais do processo de atendimento de requisições de clientes [Chen and Mohapatra 2003].

Para suprir as necessidades das atuais aplicações web, garantindo requisitos de QoS, servidores web devem se tornar providos de mecanismos e políticas de

atendimento onde clientes devem ser classificados de modo a priorizar o atendimento de suas sessões, quando deparados com sobrecargas [Cherkasova and Phaal 1999] [Cherkasova and Phaal 2002].

O modelo de servidor web com diferenciação de serviços (SWDS) [Teixeira *et al.* 2005], foi desenvolvido com preocupação de atender de maneira diferenciada as requisições de usuários e manter, ao mesmo tempo, o sistema livre de sobrecargas. Novos algoritmos de diferenciação de serviços desenvolvidos vieram somar um melhor atendimento de requisições de forma diferenciada [Traldi *et al.* 2006]. Contudo, o modelo não possui a propriedade de reconhecimento de sessões HTTP, característica importante para as aplicações atuais, que requerem garantias de que transações realizadas por meio de sessões sejam bem sucedidas. Assim, o modelo atende as requisições de forma isolada e não dá devida atenção às sessões, que são definidas como uma seqüência de requisições de um único cliente em um período de uma sessão.

Neste artigo é apresentado um esquema de reconhecimento sessões e de controle de admissão dessas sessões, incluído ao modelo de servidor SWDS, de modo que o servidor venha a fornecer maior confiabilidade no atendimento de sessões, característica fortemente requerida por aplicações atuais. Foram desenvolvidas duas políticas de controle de admissão de sessões cujo principal objetivo é atender completamente todas as sessões ativas no sistema e, para isso, rejeita novas sessões quando o sistema se apresenta sobrecarregado. Adicionalmente, esse controle ainda contribui para evitar sobrecargas no sistema. A política denominada *Hard-Threshold* se baseia estritamente no número de sessões ativas no sistema, onde um limiar é utilizado como parâmetro para a aceitação ou rejeição de uma sessão. A política de admissão de sessões Baseada em Sessão Modelo foi construída com um controle mais elaborado para admitir uma sessão no sistema. Utilizando-se de um modelo de sessão, construído por informações de um histórico de sessões, esse último controle admite ou rejeita novas sessões chegadas no sistema.

O esquema de controle de admissão de requisições é responsável por coletar informações sobre a carga de trabalho presente ao sistema e gerenciar a aceitação de novas requisições pelo servidor. Para tanto, foram criadas dois mecanismos de admissão de requisições, ambos baseados em informações sobre o tamanho das filas do *cluster*, e baseados em sessões. A diferença entre eles está no modo de rejeição de uma requisição, onde um dos mecanismos possui a característica de negociação, pela qual, busca uma requisição mais adequada para ser retirada do sistema, baseando-se nas sessões ativas.

Em sistemas de comércio eletrônico, *e-commerce*, é desejável que requisições referentes ao processo de finalização de uma compra recebam maior prioridade de atendimento do que as demais requisições. O trabalho de Cherkasova & Phaal (1999) foi o pioneiro em analisar o desempenho de servidores de comércio eletrônico considerando uma carga de trabalho baseada em sessões. Nesse trabalho os autores mostram que as sessões que resultam em vendas são maiores que sessões de não-venda, e com isso é observada a importância desse tipo de sistema em oferecer garantias de atendimento para as sessões longas. Nesse contexto, tal priorização é realizada por meio da diferenciação no atendimento de sessões, onde as requisições de sessões mais longas, e que provavelmente estão sendo finalizadas, são atendidas com maior prioridade.

Na Seção 2 a metodologia para a realização dos experimentos é apresentada. A Seção 3 descreve os mecanismos de controle de admissão de sessões *Hard-Threshold* e Baseado em Sessão Modelo (BSM). Os mecanismos de controle de admissão de requisições e as políticas de negociação desenvolvidas são apresentados na Seção 4. Na Seção 5 encontram-se os resultados obtidos dos experimentos realizados onde os CAR e CAS foram utilizados. Finalmente, na Seção 6, são descritas as conclusões.

## 2. Metodologia

Para avaliar o desempenho dos mecanismos e políticas desenvolvidos neste trabalho, foi utilizado um modelo de Rede de Filas do servidor Web com suporte a serviços diferenciados, o SWDS [Teixeira *et al.* 2005]. Os mecanismos desenvolvidos foram aplicados a este modelo de servidor web com diferenciação de serviços.

O SWDS é composto pelos módulos *Classificador*, seguido pelo *Controle de Admissão* e, finalmente, pelos *nós servidores* que constituem o *cluster*. Sua arquitetura é exibida na Figura 1. O módulo Classificador é responsável pela classificação em classes de serviços das requisições que chegam ao sistema; após a classificação, o Controle de Admissão decide pela aceitação ou não da nova requisição considerando a política vigente; finalmente, a requisição aceita é encaminhada a um dos nós servidores, o qual atende a requisição conforme o algoritmo de escalonamento ou diferenciação de serviços em vigor. Após a conclusão do processamento, os resultados são retornados ao cliente que originou a solicitação.

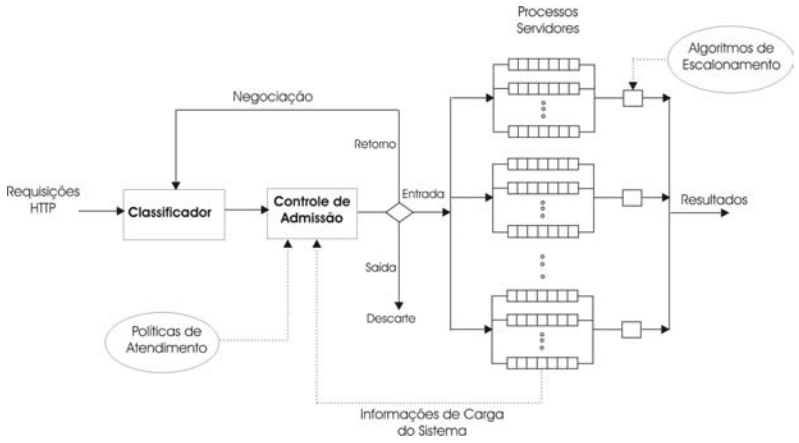


**Figura 1. Servidor Web com Diferenciação de Serviços (SWDS)**

### 2.1. Parametrização

Este trabalho seguiu a mesma parametrização utilizada em [Teixeira *et al.* 2005] cujos valores foram atribuídos com base em sistemas reais, considerando um *cluster* formado por quatro servidores Web homogêneos, onde cada nó é modelado separadamente, com sua própria CPU, disco e interface de rede. A capacidade de processamento do Classificador é definida como 8000 req./s e a do Controle de Admissão como 4000 req./s. Para o cálculo do tempo de serviço das requisições estáticas, os discos dos servidores são parametrizados com taxa de transferência de 37MBps e latência de 8,5ms, tomando-se como referência um disco IBM Deskstar 75GXP. O tempo de serviço das requisições dinâmicas é assumido como 10ms.

A simulação, feita utilizando o pacote SimPackJ [Fishwick 2004], foi a abordagem escolhida para a validação do modelo SWDS.

Para simular a carga de trabalho foram utilizados *logs* de acesso a servidores Web reais, coletados durante a Copa do Mundo 98, caracterizando uma simulação dirigida por *"traces"*. A faixa do *log* escolhida para a geração da carga foi entre 3,6 e 4,6 milhões de registros, totalizando 1 milhão de registros, usando todos os servidores do *log* do dia 11 de junho de 1998. Nessa faixa ocorre um súbito aumento da carga de trabalho, situação ideal para se avaliar a eficiência do controle de admissão de sessões desenvolvido.

Apesar desta carga de trabalho não apresentar características específicas de uma carga de um site de comércio eletrônico, onde o uso de sessões é fundamental, ela foi escolhida devido a sua grande utilização em pesquisas encontradas na literatura além da grande dificuldade de se encontrar registros de *logs* disponíveis, capazes de satisfazer os requisitos de simulações realizados neste trabalho.

Para se realizar o atendimento diferenciado de sessões, considerou-se duas classes de serviços: 1 (alta prioridade) e 0 (baixa prioridade).

### 2.2. Leitura do *Log*

Algumas modificações foram realizadas no modelo de simulação original do SWDS quando considerada a carga de trabalho baseada em sessões, uma vez que o modelo original não faz uso de informações sobre sessões presente na carga. Portanto, um campo do registro do *log* denominado clientID, foi utilizado para identificar o cliente, sendo que este dado consiste de um mapeamento do seu endereço IP. O controle das sessões é feito a partir da análise do campo clientID das requisições que chegam ao sistema, assim como em [Arlitt 2000]. Todas as requisições vindas do mesmo clientID pertencem a mesma sessão, desde que o *tempo entre chegadas dessas requisições* (*User Think Time)* [Arlitt 200] não ultrapasse o tempo limite definido pelo sistema (*timeout*), configurado em 100 segundos para este trabalho.

O *User Think Time* médio obtido pelo estudo de Arlitt (2000) para esta carga, foi 63 segundos, porém nos resultados obtidos por simulações com este valor para o parâmetro *timeout*, não foi possível reproduzir sobrecargas ao sistema, de modo que não haveria possibilidades de avaliar os controles desenvolvidos. Outros experimentos foram realizados variando-se o valor deste parâmetro e após os resultados obtidos o valor de 100 segundos foi escolhido, quando observou-se maior sobrecarga ao sistema.

Uma sessão é considerada como finalizada quando todas suas requisições são atendidas, sendo que neste trabalho uma sessão expirada também foi considerada como uma sessão finalizada.

As requisições aceitas no sistema são enviadas aos nós do *cluster* conforme o gerenciamento feito pelo algoritmo de escalonamento *Shortest Queue First* (SQF), portanto, requisições de uma mesma sessão podem ser atendidas por diferentes servidores do *cluster*.

### 2.3. Classificação das Requisições

No modelo SWDS o módulo Classificador classificava as requisições de modo aleatório. Para este trabalho, houve a necessidade de se realizar a classificação

considerando-se as sessões. Para tanto, todas as requisições de uma determinada sessão foram classificadas com a mesma prioridade, porém, a classificação da sessão continuou sendo feita de forma aleatória.

### 2.4. Métricas de Desempenho

As métricas consideradas neste trabalho foram o tamanho das filas dos servidores do *cluster* e a quantidade de sessões e requisições atendidas pelo sistema. Com estas informações adquiridas de simulações realizadas, permitiu-se analisar o desempenho de cada mecanismo e políticas desenvolvidos.

## 3. Controle de Admissão de Sessões (CAS)

O controle de admissão de sessões desenvolvido, além de auxiliar a manter a carga do sistema em níveis aceitáveis, tem como maior aspiração atender completamente todas as sessões que se tornaram ativas no sistema. Para ajudar a manter a carga do sistema dentro de níveis aceitáveis, esse controle aceita ou rejeita uma nova sessão conforme o estado em que o servidor se encontra. Caso a carga do sistema, relacionada ao número de sessões ativas, esteja em níveis inaceitáveis, o controle deverá rejeitar uma nova sessão.

Com esse objetivo, foram desenvolvidos métodos para prever se o sistema suporta o atendimento de uma nova sessão por completo, ou seja, o atendimento de todas requisições que a compõem.

### 3.1. Arquitetura

A arquitetura do controle de admissão de sessões desenvolvido e incorporado ao SWDS pode ser visto na Figura 2. Este mecanismo tem a função de gerenciar a aceitação de novas sessões ao sistema, evitando que sessões sejam posteriormente canceladas quando há sobrecarga ao sistema. Destacam-se os seguintes componentes:

***Buffer de Sessões Ativas:*** é o *buffer* utilizado para limitar o número de sessões que podem estar ativas no sistema. O limite especificado para este trabalho foi a metade da soma da capacidade das filas do *cluster*, o que permite que no mínimo duas requisições de cada sessão ativa no sistema sejam atendidas, em um cenário onde todas sessões emitem requisições ao sistema.

***Histórico de Sessões****:* onde são armazenadas informações sobre a quantidade de requisições pertencentes às sessões finalizadas no sistema. O número de sessões armazenadas nesse *buffer* pode ser configurado. Quanto menor o número de sessões, mais freqüente é a atualização do *buffer* e, conseqüentemente, mais suscetível a mudanças pontuais da carga o controle se torna, pois, ao se apresentar cheio, as sessões finalizadas mais recentemente tomam o lugar das mais antigas nesse histórico.

***Mecanismo de Controle:*** onde são utilizadas as políticas de controle de admissão de sessões que serão apresentadas a seguir.

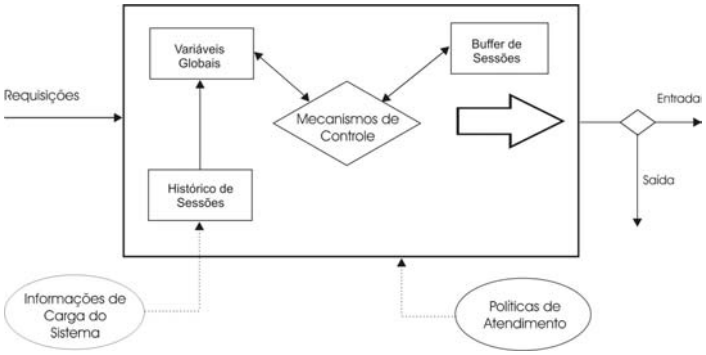


**Figura 2. Arquitetura do Controle de Admissão de Sessões**

Foram desenvolvidas duas políticas de controle de admissão de sessões; *Hard-Threshold* e Baseada em Sessão Modelo (BSM). Ambas utilizam informações do *buffer* de sessões para rejeitar sessões quando há sobrecarga no sistema. No entanto, uma delas utiliza informações do histórico de sessões para construir uma sessão modelo, na qual se baseia para prever se o sistema será capaz de atendê-la completamente. Com isso o sistema se torna apto a prevenir possíveis cancelamentos de sessões e não somente fazer rejeições de sessões quando o sistema já se apresenta sobrecarregado.

### 3.2. Política *Hard-Threshold*

A política *Hard-Threshold* utilizada no CAS baseia-se em uma quantidade máxima de sessões, configurada para tomar decisões sobre aceitar ou rejeitar uma sessão chegada ao sistema. Considerou-se que a fila de um servidor comporta 1.024 requisições, e que o *cluster* de servidores é formado por quatro servidores, portanto, permite-se que 2.048 sessões estejam ativas no sistema ao mesmo tempo, ou seja, o *buffer* de sessões comporta esta quantidade de sessões.

O algoritmo utilizado pela política *Hard-Threshold* verifica a quantidade de sessões ativas no *buffer* de sessões, para cada nova sessão chegada ao sistema. Nesse momento o algoritmo toma decisões de aceitar ou rejeitar a sessão. Caso o *buffer* tenha atingido seu limite, a sessão será rejeitada. Caso contrário, a sessão será aceita no sistema e sua identificação será armazenada no *buffer* de sessões.

Esse algoritmo é restrito a uma quantidade máxima de sessões no sistema e, portanto, toma a decisão de rejeitar sessões somente se o sistema se apresentar sobrecarregado, não fazendo nenhum tipo de estimativa para saber se o sistema será capaz de concluir a sessão após sua aceitação.

### 3.3. Política Baseada em Sessão Modelo (BSM)

Essa política é empregada no Controle de Admissão de Sessões (CAS) e fundamenta-se em informações provindas de uma sessão modelo para tomar decisões de rejeição de novas sessões chegadas ao sistema. O principal objetivo dessa política é permitir que uma nova sessão seja aceita no sistema somente se sua aceitação não prejudicar as sessões que já se apresentam ativas e se for previsto que uma vez iniciada a sessão, o sistema tem grandes possibilidades de finalizar a nova sessão, atendendo todas suas requisições com um mínimo de confiabilidade.

O mecanismo da política atua do seguinte modo: mediante a requisição de uma nova sessão, o sistema realiza uma previsão da quantidade de requisições que estão

sendo esperadas pelas sessões ativas e afere o impacto da aceitação de uma nova sessão na carga do sistema. Caso seja previsto que a admissão da nova sessão trará uma sobrecarga ao sistema, a sessão não será aceita pelo controle.

Após essa verificação, a política faz mais um controle caso a sessão tenha sido aceita pela previsão. O controle obtém dados sobre o tamanho do *buffer* de sessões e caso este esteja em seu limite, a sessão não poderá ser aceita. Essa situação pode acontecer no caso em que a maior parte de sessões ativas estão sendo finalizadas, sendo que a distribuição modelo indicará que espera por uma quantidade pequena de requisições ao sistema, mas por outro lado, a quantidade de sessões ativas é grande.

Para prever a quantidade de requisições esperadas, o controle utiliza informações de uma Distribuição Modelo, ou Sessão Modelo já que ela caracteriza o comportamento de uma sessão, com base em um histórico de sessões finalizadas no sistema. O controle verifica a aceitação de cada nova sessão chegada ao sistema. Para isso é feita uma estimativa da quantidade de requisições que poderão chegar ao sistema nos próximos *Tmax* segundos, onde *Tmax* é o tempo máximo de uma sessão parametrizado no sistema. Caso uma sessão ultrapasse este tempo máximo, o sistema a considera como uma sessão expirada, no entanto, para os resultados obtidos neste trabalho, esta sessão foi contabilizada como uma sessão finalizada.

Considera-se que $R_{Tmax}$ é a quantidade de requisições previstas das sessões ativas para chegarem ao sistema nos próximos *Tmax* segundos, $R_{NovaSessão}$ é a quantidade de requisições previstas para uma nova sessão para os próximos *Tmax* segundos e $R_{Filas}$ é o total de requisições presentes nas filas dos servidores do *cluster* que esperam por atendimento.

Considera-se também que *TH* é o *throughput* de requisições, ou seja, é a quantidade de requisições atendidas por segundo e $T_{MédioSessão}$ é o tempo médio de uma sessão obtido pelo cálculo da média aritmética dos tempos decorridos de todas sessões ativas no sistema. Optou-se por multiplicar o *throughput* do sistema pelo tempo médio de sessão ($T_{MédioSessão}$) em vez de multiplicá-lo pelo tempo máximo de uma sessão (*Tmax*) para permitir que a estimativa da quantidade de requisições a ser atendida se tornasse mais adequada à carga presente no sistema.

A capacidade do sistema, *Cmax*, é a quantidade máxima de requisições que poderão estar presente no sistema em determinado momento, sendo obtido pela soma dos tamanhos máximo das filas dos nós do *cluster*. Portanto, para que o sistema seja considerado capaz de atender uma nova sessão, a Equação 1 deve ser satisfeita, ou seja, a quantidade de requisições previstas para chegarem no sistema nos próximos *Tmax* segundos deve ser inferior a capacidade de atendimento do sistema. Caso contrário, a sessão não será aceita no sistema sendo então descartada.

$$R_{Tmax} + R_{NovaSessão} + R_{Filas} - TH * T_{MédioSessão} < Cmax \qquad (1)$$

### 3.3.1. Distribuição Modelo

O problema, portanto, consiste em estimar a quantidade de requisições que deverão chegar nos próximos *Tmax* segundos de todas sessões ativas no sistema. Para isso, propõe-se a adoção de uma Distribuição Modelo.

A Distribuição Modelo, ou Sessão Modelo, é uma distribuição no tempo do número de requisições a serem enviadas ao sistema por uma sessão. Para a construção

da Distribuição Modelo, é estabelecido um tempo máximo para a duração de uma sessão e um número de intervalos para a discretização do tempo. Nos experimentos realizados neste trabalho, esses parâmetros foram configurados da seguinte maneira:

Tempo Máximo de uma sessão (*Tmax*) = 900 segundos

Número de Intervalos *(I)*= 100

A configuração do tempo máximo de uma sessão foi baseada no maior tempo de sessão encontrado no experimento de referência, que foi de 1.726 segundos sendo, portanto, configurado como aproximadamente a sua metade. O número de intervalos foi estabelecido aleatoriamente. Com esses valores, a Discretização do Tempo (*D*) é calculada pela Equação 2, pois seu valor é obtido pelo resultado da divisão do tempo máximo de uma sessão pelo número de intervalos desse tempo.

$$D = Tmax / I \qquad (2)$$

Portanto, o período de duração de uma sessão, 900 segundos, foi dividido em 100 intervalos de 9 segundos cada. Para cada requisição de uma sessão, deve ser calculado em qual intervalo de tempo da sessão ela se encontra.

Suponha que uma requisição chegue ao sistema no Tempo = 31 de uma sessão. Considera-se, portanto, que a requisição da sessão se encontra no 3º Intervalo (*timeslot*) do tempo discretizado, pois 31/9 = 3,4 aproximadamente.

No sistema, a distribuição modelo é freqüentemente atualizada. Para isso, foi construído um histórico das últimas *N* sessões atendidas com sucesso pelo sistema. Quando uma sessão é finalizada, ela é inserida no histórico, ou seja, a quantidade de requisições por *timeslot* do tempo discretizado, desta sessão, é armazenada, e após isso, a distribuição modelo atualizada. Este processo mantem o sistema ajustado ao comportamento das sessões. Para as simulações realizadas, o parâmetro *N* foi configurado com o valor de 5.000 sessões. Quanto menor esse valor, mais freqüente é a atualização do *buffer* e, conseqüentemente, mais assídua é a atualização da distribuição modelo, portanto, o controle torna-se propenso a ser suscetível às variações da carga.

O gráfico da Figura 3 exibe o modelo da distribuição obtida no final do experimento onde essa política foi aplicada. A partir desses gráficos, pôde-se concluir que a sessão modelo obtida atende ao comportamento visto no estudo sobre a caracterização de sessões no *log* da Copa de 98 [Arlitt 2000], onde a maior parte das requisições de uma sessão são enviadas ao servidor logo em seu início.

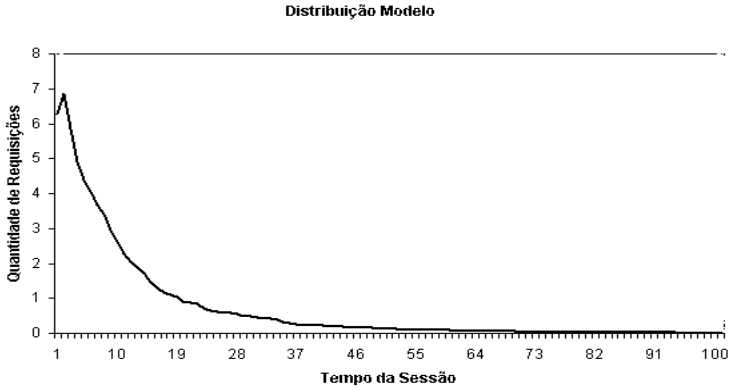


**Figura 3. Sessão Modelo**

De posse de uma distribuição que estima o comportamento de uma sessão quanto à quantidade de requisições no tempo, é possível prever quantas requisições deverão ser enviadas por uma sessão em um instante qualquer.

Portanto, é necessário conhecer $R_{Tmax}$, número de requisições estimadas para os próximos *Tmax* segundos para todas as sessões que estão sendo atendidas pelo sistema. Para isso, faz-se o seguinte cálculo:

$$R_{Tmax} = \sum_{i=1}^{n} r_{iTmax} \qquad (3)$$

onde *n* é o número de sessões ativas no sistema e $r_{iTmax}$ é o número de requisições estimadas para uma sessão *i* nos próximos *Tmax* segundos. Com $R_{Tmax}$ calculado, a Equação 1 utilizada para determinar se uma nova sessão será aceita no sistema, é facilmente resolvida.

### 3.3.2. Estudo da Política

Para verificar o comportamento da política baseada em sessão modelo desenvolvida, foram realizados experimentos variando-se a quantidade de sessões permitidas para ocupar o histórico de sessões (*N*). Utilizou-se como parametrização um *timeout* de 900 segundos, equiparando ao tempo máximo estipulado para uma sessão (*Tmax*). Com este parâmetro, melhores resultados puderam ser obtidos, uma vez que sessões mais longas puderam ser geradas.

O gráfico mostrado na Figura 4 exibe os resultados obtidos desses experimentos. Verifica-se que quanto menor a quantidade de sessões permitidas no histórico, mais sessões foram atendidas pelo sistema, já que menor quantidade de sessões foram rejeitadas pelo CAS.

Se um conjunto de sessões longas prevalecer no histórico, uma nova sessão curta (poucas requisições) será tratada como longa (muitas requisições). Após o seu término, ela será considerada no histórico, fazendo com que a distribuição modelo se adeqüe a nova sessão mais curta. Se o maior conjunto de sessões no histórico é de sessões curtas, uma nova sessão longa será tratada como curta e favorecida nesse modelo. Portanto, existirá uma interdependência entre o conjunto real do *log* de entrada e o número de sessões que é mantido no histórico para garantir adaptação.

Certamente, um valor para *N* menor favorece uma adaptação mais rápida, no entanto, como é feito somente uma estimativa do modelo de uma nova sessão por esse esquema, não é fornecida a garantia de que uma nova sessão será atendida, já que uma nova sessão longa, em uma situação onde foi aceita com base em uma sessão modelo gerada por sessões caracterizadas como curtas, poderá não ser atendida completamente, pois haverá maiores chances de o sistema se tornar sobrecarregado e não ser capaz atendê-la.

Para todos os experimentos os tamanhos das filas atingiram o mesmo tamanho de 1.906 requisições. Com isso pode-se concluir que quanto menos sessões armazenadas no histórico, mais dinâmico o sistema se torna, se adaptando mais rapidamente a situação da carga.


**Figura 4. Quantidade de sessões atendidas com a variação de quantidade de sessões no histórico de sessões**

### 3.4. Resultados Experimentais

O principal objetivo dos experimentos realizados é verificar a eficiência das políticas propostas empregadas no controle de admissão de sessões, observando as vantagens advindas para o desempenho do servidor SWDS em relação ao atendimento de sessões. Em todos os experimentos realizados nesta seção são utilizados os parâmetros descritos na Seção 2.

Nos experimentos apresentados nesta seção, nenhum controle de admissão de requisições foi utilizado, e também, os tamanhos das filas do *cluster* não foram restringidas, de modo que se permitisse observar a eficiência do controle de admissão relacionada às sobrecargas do sistema.

A Tabela 1 exibe os resultados relativos às sessões, obtidos de simulações onde o Controle de Admissão de Sessões (CAS) foi utilizado com as políticas *Hard-Threshold* e BSM empregadas. Pelos resultados obtidos, verifica-se que a quantidade de sessões admitidas por ambas políticas foi de 11.520 sessões, e portanto, a mesma quantidade de sessões foram rejeitadas pelas políticas. No entanto, o modo como as sessões foram rejeitadas é distinto. Com o emprego da política *Hard-Threshold* foram rejeitadas 533 sessões, todas com base em informações do *buffer* de sessões. Ao ser utilizada a política BSM, observa-se que 244 sessões foram rejeitadas com base nestas informações do *buffer* de sessões, e 289 rejeitadas por estimativas de atendimento feitas por informações da distribuição modelo.

**Tabela 1. Resultados sobre sessões para os mecanismos de controle de admissão de sessões**

| Resultados sobre Sessões | *Hard-Threshold* | BSM |
|---|---|---|
| Chegadas | 12.053 | 12.053 |
| Rejeições pelo CAS | 533 | 533 |
| - Limite do *buffer* | 533 | 244 |
| - Estimativa de atendimento | 0 | 289 |
| Admissões | 11.520 | 11.520 |
| Términos | 10.377 | 10.379 |
| *Throughput* (Sessões/seg) | 5,484672304 | 5,485729387 |

Apesar de ter a mesma quantidade de sessões rejeitadas, foram obtidos resultados diferentes quanto o atendimento de requisições para as políticas

desenvolvidas, como pode ser visto na Tabela 2. Com a política *Hard-Threshold* o número de requisições chegadas ao sistema foi menor do que para a política BSM. É provável que o mecanismo de previsão de atendimento de uma nova sessão tenha gerado este efeito na carga de trabalho. Com isso, permite-se concluir que com a política BSM foram atendidas sessões maiores, ou seja, com maior número de requisições.

**Tabela 2. Resultados sobre requisições para os mecanismos de controle de admissão de sessões**

| Resultados sobre Requisições | *Hard-Threshold* | BSM |
|---|---|---|
| Chegadas | 714.734 | 717.163 |
| Rejeições pelo CAR | 0 | 0 |
| Descartes | 0 | 0 |
| Admissões | 714.734 | 717.163 |
| Retiradas das filas | 0 | 0 |
| Términos | 714.620 | 717.049 |
| *Throughput* (Req/seg) | 377,7061311 | 378,9899577 |

Observa-se por resultados da Tabela 3 que o sistema se apresentou sobrecarregado para ambas políticas utilizadas, já que suas filas atingiram uma quantidade máxima superior a 1.024, parâmetro baseado em servidores reais para o tamanho de fila padrão utilizado. Esta situação deve ser contornada pelo mecanismo de controle de admissão de requisições, explicado na próxima seção. No entanto, comparando-se com o resultado obtido do experimento onde nenhum controle admissão de sessões foi utilizado, o sistema se tornou menos sobrecarregado com o uso das políticas.

**Tabela 3. Maior tamanho das filas para os mecanismos de admissão de sessões**

| Filas dos Servidores | Sem CAS | *Hard-Threshold* | BSM |
|---|---|---|---|
| Servidor 1 | 6.135 | 2.618 | 1.908 |
| Servidor 2 | 6.135 | 2.618 | 1.908 |
| Servidor 3 | 6.135 | 2.618 | 1.908 |
| Servidor 4 | 6.136 | 2.619 | 1.909 |

### 3.5. Considerações

De modo geral, as políticas tiveram um desempenho semelhante, no entanto a Baseada em Sessão Modelo (BSM) obteve vantagem sobre a *Hard-Threshold*, devido à quantidade de requisições atendidas, uma vez que fez prevalecer o atendimento de sessões mais longas, característica marcante de sessões *e-business*. Além disso, a política BSM ainda conseguiu manter o sistema menos sobrecarregado do que para a outra política utilizada.

Enfim, esses métodos contribuíram para reduzir a possibilidade de que uma sessão venha a ser cancelada futuramente, já que há maiores possibilidades de que uma requisição pertencente a uma sessão aceita sem a verificação do estado do sistema, seja rejeitada quando este se apresentar sobrecarregado.

## 4. Controle de Admissão de Requisições (CAR)

A diferenciação de serviços para sessões se dá por meio da classificação de suas requisições. Para isso todas as requisições pertencentes a uma sessão são classificadas com a mesma prioridade, tornando a sessão também classificada. Esta atribuição de classes às requisições e, portanto, às sessões, é realizada no módulo Classificador de Requisições do SWDS. A diferenciação no atendimento de requisições, no entanto, é feita em parte, no módulo de Controle de Admissão de Requisições (CAR) através de políticas de atendimento às requisições. Outra função, e a principal deste módulo, é manter o sistema livre de sobrecargas e, para isso, requisições são rejeitadas utilizando-se esse controle.

Para atender as necessidades da nova característica de reconhecimento de sessões, foram criadas dois novos mecanismos de admissão de requisições, adaptados ao atendimento de sessões, o mecanismo de Admissão por Tamanho de Fila Sem Negociação e o mecanismo de Admissão por Tamanho de Fila Com Negociação. Os dois mecanismos desenvolvidos apóiam-se na informação sobre o tamanho das filas do *cluster*. Caso as filas se apresentem em seu limite de capacidade, requisições devem ser rejeitadas e, como conseqüência, a sessão a qual elas pertencem deve ser cancelada, controlando com isso a sobrecarga ao sistema.

O mecanismo de Admissão por Tamanho de Fila Com Negociação utiliza-se de políticas de negociação de forma a priorizar o atendimento de requisições e consequentemente de suas sessões. A priorização no atendimento de requisições se faz de grande importância, já que em um cenário de um web site de comércio eletrônico, por exemplo, se considerarmos como se a última requisição de uma sessão enviada ao servidor fosse a confirmação de uma compra por um cliente, e caso esta requisição não pudesse ser processada, todo procedimento da compra poderia ser perdido, já que a sessão seria cancelada para a requisição rejeitada. Enquanto isso, uma outra sessão em que o cliente estivesse somente passeando pelo site, ou seja, de menor prioridade de atendimento, continuaria sendo atendida. Logicamente, o atendimento da sessão no primeiro caso se mostra de maior importância, e neste cenário, portanto, observa-se a importância de se utilizar mecanismos de controle da admissão de requisições com políticas que se preocupam em priorizar o atendimento das sessões.

### 4.1. Admissão Segundo o Tamanho de Fila Sem Negociação

Este mecanismo de admissão de requisições utiliza-se de um política que estabelece um tamanho máximo para as filas dos servidores do *cluster*. Caso uma requisição chegue ao sistema e encontre todas as filas com este limiar alcançado, ela será sumariamente recusada, independente de sua classe e de qualquer tipo de informações pertinentes a sua sessão. Essa abordagem é semelhante à encontrada no servidor web Apache, com a qual, novas requisições HTTP são rejeitadas caso o tamanho de sua fila ultrapasse 1.024 clientes. Em nível de rede esta situação também ocorre, onde pacotes mais novos são rejeitados por dispositivos roteadores.

Com essa política, quando há uma sobrecarga no sistema, uma nova requisição chegada ao sistema é rejeitada pelo controle de admissão de requisições e sua respectiva sessão deve ser descartada. Portanto, a sessão descartada será sempre a da requisição que chegou ao sistema, para a qual seu atendimento foi negado.

O descarte de uma sessão implica, de forma geral, em inutilizar todo o processamento já realizado a esta sessão. O pior caso, portanto, é quando uma sessão está preste a ser finalizada e sua última requisição é rejeitada. Caso houver ainda alguma requisição dessa sessão presente nas filas do *cluster*, ela deve ser excluída, pois seu processamento seria inútil após o cancelamento de sua sessão.

### 4.2. Admissão Segundo o Tamanho de Fila Com Negociação

Este mecanismo de controle de admissão utiliza políticas de negociação para escolher a sessão que deverá ser cancelada quando uma requisição for rejeitada. Ao encontrar uma sessão ativa no sistema que satisfaça aos requisitos de descarte da política, esta será cancelada no lugar da sessão da requisição rejeitada. A requisição é então atendida pelo sistema e todas as requisições da sessão cancelada que estiverem nas filas dos servidores, serão descartadas. Portanto, a sessão descartada será uma sessão escolhida pelo sistema conforme a política de negociação utilizada.

Foram desenvolvidas quatro políticas de negociação de sessões que tomam decisões com base na quantidade de requisições por sessão, no tempo das sessão ou na prioridade das sessões.

***Negociação por descarte de sessão mais recente*** - Esta política de negociação de sessão tem como principal objetivo a preservação de sessões ativas que estão há mais tempo no sistema, as quais possuem maior probabilidade de finalização. Quando uma requisição chega ao sistema e este se encontra sobrecarregado, este mecanismo busca por uma sessão no *Buffer* de Sessões Ativas que esteja a menos tempo ativa no sistema, para ser cancelada. Para não desperdiçar serviços dos recursos do sistema, todas as requisições pertencentes a essa sessão que se encontram nas filas dos servidores, ou em qualquer outro recurso, são excluídas.

***Negociação por descarte de sessão mais recente e de menor duração*** - Esta política além de preservar as sessões ativas mais antigas no sistema, como a apresentada no item anterior, ainda se preocupa em preservar as sessões que estiveram ativas por mais tempo no sistema, já que teoricamente estas sessões têm maiores probabilidades de finalização. O mecanismo busca a sessão mais recente no sistema dentre as sessões de menor tempo de duração.

A duração de uma sessão é considerada como o tempo entre a primeira e a última requisição emitida por essa sessão. Portanto, uma sessão mais antiga no sistema que tenha seu período de duração menor do que uma sessão mais recente no sistema deve ser cancelada por esta política.

***Negociação por descarte de sessão mais recente, de menor tempo de duração e de classe baixa*** - Esta política de negociação preza por atender sessões de maior prioridade e ao mesmo tempo por fazer com que sessões que se encontram há mais tempo ativa no sistema continuam sendo atendidas. Neste caso, ao utilizar esta política, o modelo irá priorizar o cancelamento de sessões cuja classe será de menor prioridade. Para isso, as classes atribuídas às sessões pelo Classificador do SWDS serão verificadas. Neste trabalho, as sessões foram classificadas em classes 0 e 1, menor e maior prioridade, respectivamente.

***Negociação por descarte de sessão mais recente e de menor tamanho*** - Com a informação da quantidade de requisições emitidas por cada sessão ativa no sistema, esta

política toma decisões de qual sessão deve ser cancelada, caso uma requisição seja rejeitada quando o sistema se encontra sobrecarregado. O sistema irá cancelar a sessão mais recentes dentre aquelas de menor tamanho (total de requisições emitidas por uma sessão). Esta política além de preservar as sessões ativas mais antigas no sistema, ainda se preocupa em atender as sessões que tiveram mais requisições processadas pelos servidores do *cluster*, caso ocorra uma sobrecarga no sistema. Deste modo, com a utilização desta política, evita-se desperdiçar todo o processamento já realizado para as requisições das sessões que poderão ser canceladas.

### 4.3. Resultados Experimentais

Os gráficos a seguir exibem os resultados obtidos de experimentos, nos quais foram utilizados os mecanismos de controle de admissão de requisição baseado no tamanho das filas do *cluster*, sem utilizar políticas de negociação e utilizando políticas de negociação, para a escolha de qual requisição deve ser descartada. Não foi utilizado o CAS para se obter resultados especificamente relacionados ao atendimento das requisições. Ao ser rejeitada uma requisição pelo controle de admissão de requisições, sua sessão é cancelada e todas as requisições pertencentes à sessão cancelada que estiverem na fila do *cluster* são excluídas.

A Figura 5 nos mostra a quantidade de sessões atendidas para os mecanismos de controle de admissão de requisições. O mecanismo de CAR por Tamanho de Fila e Sem Negociação é caracterizado pela descrição “Sem Negociação”, e para o mecanismo de CAR por Tamanho de Fila e Com Negociação são exibidos os resultados obtidos para cada política de negociação por ele utilizado, onde os nomes descritos as caracterizam.

Apesar de o mecanismo de CAR por Tamanho de Fila Sem Negociação ter permitido o atendimento de maior quantidade de sessões em relação ao mecanismo de CAR por Tamanho de Fila Com Negociação para todas políticas de negociações, a quantidade de requisições atendidas foi menor (Figura 6).

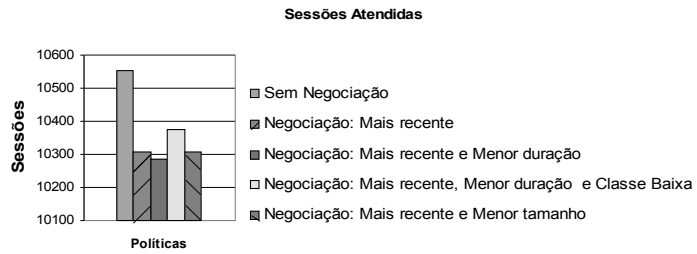


**Figura 5. Quantidade de sessões atendidas pelos mecanismos de CAR**

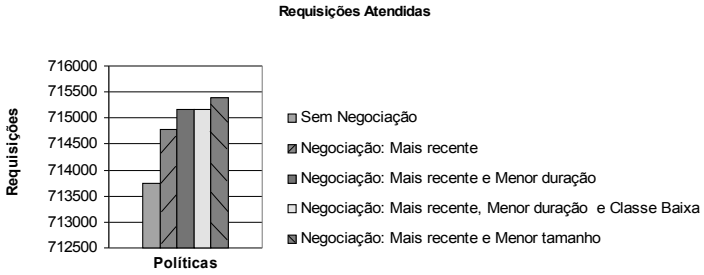


**Figura 6. Quantidade de requisições atendidas pelos mecanismos de CAR**

Portanto, em relação ao seu principal objetivo, gerenciar a admissão de requisições no sistema, o mecanismo de CAR por Tamanho de Fila Sem Negociação se mostrou menos eficiente, já que acabou rejeitando mais requisições do que para o mecanismo que utiliza políticas de negociação. Este resultado pode ainda ser comprovado pelos gráficos das Figuras 8 e 9 que exibem a quantidade de requisições canceladas pelos mecanismos de CAR e a quantidade de requisições retiradas das filas, respectivamente. As requisições são retiradas das filas como conseqüência do cancelamento de requisições das sessões escolhidas pela política de negociação vigente para ser cancelada no lugar da sessão da requisição rejeitada pelo CAR. Portanto, a soma desses dois valores representa a real quantidade de requisições rejeitadas pelo sistema.

Observa-se pelo gráfico da Figura 5 que a maior parte das sessões atendidas pelo mecanismo de CAR por Tamanho de Fila Sem Negociação eram as que tinham poucas requisições. Portanto, sessões maiores, ou seja, com maior quantidade de requisições, puderam ser canceladas e todo o processamento de requisições anteriormente atendidas foi desperdiçado. Assim, a quantidade de requisições que são retiradas das filas do *cluster* para o mecanismo que não utiliza políticas de negociação se mostrou maior do que o dobro para o mecanismo com negociação, como pode ser observado nos gráficos das Figuras 8 e 9.

Dentre as políticas de negociação, observa-se na Figura 5 que com a utilização da política de descarte de sessões de menor duração pelo CAR, o sistema atendeu menos sessões do que com as outras políticas de negociação empregadas. Com isso, permite-se concluir que uma sessão de maior duração poderia ocupar uma posição de sessão ativa no sistema desnecessariamente, provocando a rejeição de novas sessões quando a quantidade máxima de sessões ativas no sistema fosse alcançada.

Pode-se considerar também, que uma sessão de maior duração não necessariamente emite mais requisições ao sistema do que uma sessão de menor duração, para a carga de trabalho utilizada, já que com o emprego da política de descarte de sessões de menor tamanho, o sistema conseguiu atender mais requisições do que com a de menor duração.

Ao ser utilizada a política de descarte de sessões mais recentes no CAR, o sistema atendeu menos requisições do que quando empregadas outras políticas, exibido no gráfico da Figura 6. Com base neste resultado, pode-se considerar para a maioria das sessões desta carga, foram emitidas mais requisições ao sistema logo no início de seus períodos de duração.

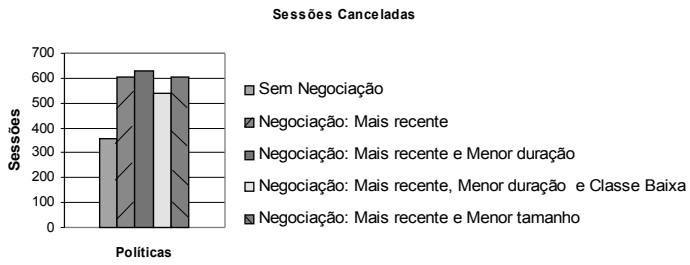


**Figura 7. Quantidade de sessões canceladas pelos mecanismos de CAR**

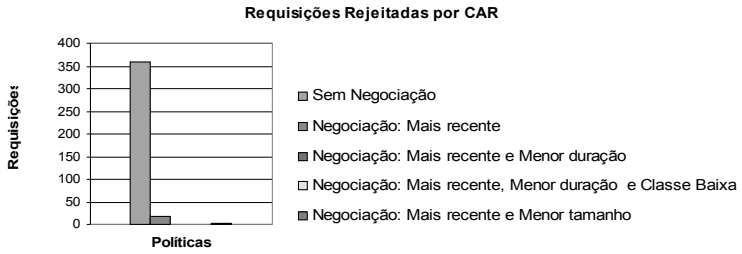


**Figura 8. Quantidade de requisições rejeitadas pelos mecanismos de CAR**

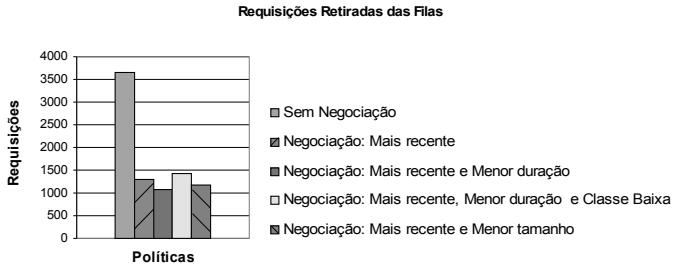


**Figura 9. Quantidade de requisições retiradas das filas pelos mecanismos de CAR**

Com o emprego da política de negociação de descarte de sessão mais nova, de menor duração e de classe baixa, a quantidade de requisições atendidas foi ainda maior do que com a utilização da política de descarte de sessão mais nova e de menor duração. A classificação de sessões foi aleatória, portanto, coincidentemente as sessões de classe 0, escolhidas para serem canceladas, haviam enviado ao sistema menos requisições do que as de classe 1. Entretanto, todas as sessões de classe 1 (maior prioridade) chegadas ao sistema, foram atendidas, com a utilização do algoritmo de negociação que procura descartar sessões de classe baixa, ou seja, atendendo ao seu principal objetivo, como pode ser observado na Tabela 4 de classes de sessões.

**Tabela 4. Resultado sobre classe de sessões utilizando a política de negociação de Sessão mais recente, de menor duração e de classe baixa**

| Resultados sobre Classe de Sessões | Classe 1 | Classe 0 |
|---|---|---|
| Chegadas | 715.979 | 716.695 |
| Admissões (0/1) | 5.941 | 6.112 |
| Canceladas (0/1) | **0** | **538** |
| Términos (0/1) | 5.941 | 5.574 |
| *Throughput* (0/1) | 3,140063 | 2,946089 |

Em relação às filas dos servidores, para todos os experimentos onde as políticas de negociação foram utilizadas, as filas mantiveram-se abaixo do limite configurado (1.024 requisições).

## 4.4. Considerações

Avaliando-se os resultados obtidos por meio de simulações, permite-se afirmar que o mecanismo de CAR por Tamanho de Fila Sem Negociação se mostra inflexível quanto à escolha de descarte de requisições, sem existir qualquer opção de escolha de qual

sessão deve ser cancelada caso uma requisição seja rejeitada pelo sistema. Essa situação pode ser contornada com as políticas de negociação desenvolvidas.

Concluí-se também, que os dois mecanismos de CAR desenvolvidos, atenderam seus objetivos. Porém, o mecanismo de CAR por Tamanho de Fila Com Negociação passa a ser melhor empregado quando se tem a preocupação de se evitar cancelar uma sessão com determinadas características que tornam seu atendimento prioritário dentre as demais sessões. Neste trabalho, portanto, deu-se grande importância ao atendimento às sessões mais antigas no sistema, característica de sessões em sites *e-commerce*.

## 5. CAS e CAR

Nesta seção serão exibidos os resultados obtidos de experimentos onde foram utilizados mecanismos de CAS e CAR em conjunto. Nas figuras a seguir são mostrados os valores relativos ao atendimento de sessões e requisições, bem como os de cancelamento de sessões. Em cada experimento, tem-se como parâmetro de comparação, resultados de simulações onde o controle de admissão de requisições não foi utilizado.

A Figura 10 exibe a quantidade de sessões atendidas para as políticas de CAS desenvolvidas. Com a utilização da política BSM no CAS, foram rejeitadas mais sessões do que com a política *Hard-Threshold*, ambas aplicadas em conjunto com as políticas de CAR, valores exibidos na Figura 11. Com a rejeição de maior quantidade de sessões, menos sessões foram canceladas pelo sistema para a política BSM, pois a previsão da capacidade de atendimento de sessões pelo sistema assim permitiu, como pode ser observado na Figura 12. Observa-se também, que mais requisições foram atendidas com a utilização da política BSM no CAS para quase todos os algoritmos de CAR, na Figura 13.

De posse desses resultados, permite-se concluir que o CAS com a política BSM teve um ligeiro melhor desempenho, devido ao atendimento de maior quantidade de requisições pelo sistema, com quase a mesma quantidade de sessões completadas. Isso se deve ao fato de que o CAR permitiu que sessões mais longas pudessem ser atendidas com um maior número de rejeições de novas sessões.

O CAS utilizando a política BSM, atendeu a um maior número de sessões quando simulado em conjunto com o CAR por TF Baseado em Sessões, cuja política de negociação para descarte de sessões utilizada, foi a de menor duração. Porém, o CAS que segue a política *Hard-Threshold* para descarte de sessões, atende a mais requisições, mesmo para maior quantidade de sessões canceladas pelo CAR. Por isso, visto que o CAS com a política *Hard-Threshold* não rejeitou sessão alguma (Figura 11), permite-se afirmar que o CAR cumpriu seu objetivo de atendimento de sessões mais longas, já que para um menor número de sessões atendidas foi obtido maior quantidade de requisições atendidas. Para o CAS com a política BSM, apesar de ter atendido maior quantidade de sessões, a baixa quantidade de requisições atendida permite concluir que as sessões rejeitadas pelo CAS foram aquelas que tinham maior número de requisições.

As simulações onde o CAR com BSM foi utilizado com a política de negociação de descarte de sessões que considera o atendimento prioritário de sessões de classe de alta prioridade, têm como resultado que o número de sessões atendidas com o CAS utilizando a política de admissão de sessões Baseado em Sessão Modelo apresenta uma discreta vantagem no atendimento de sessões e requisições.

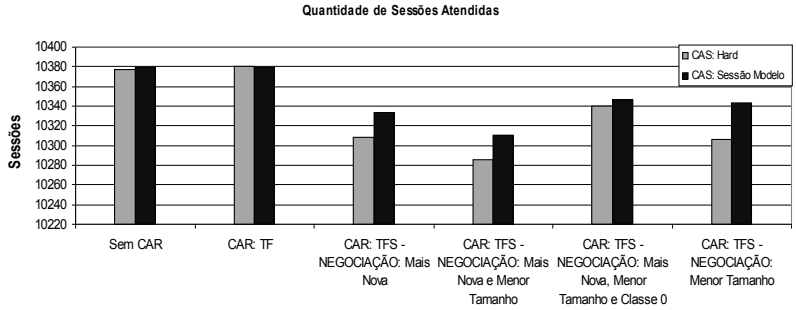


**Figura 10. Quantidade de sessões atendidas pelas políticas de CAS utilizadas em conjunto com os mecanismos de CAR**

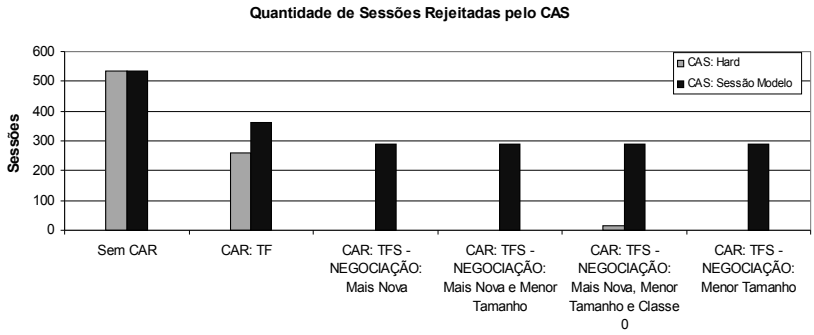


**Figura 11. Quantidade de sessões rejeitadas pelas políticas de CAS utilizadas em conjunto com os mecanismos de CAR**

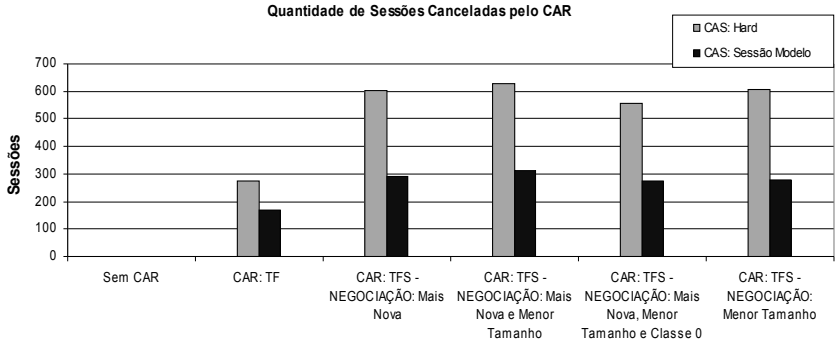


**Figura 12. Quantidade de sessões canceladas pelos mecanismos de CAR utilizados em conjunto com as políticas de CAS**

Com base nos resultados obtidos, pode-se concluir que as sessões rejeitadas pelo CAS com BSM não foram as que possuíam maior quantidade de requisições em um cenário diferente do apresentado anteriormente, com a prioridade de atendimento de sessões de classe de alta prioridade e não somente daquelas de menor duração. Com isso, menos sessões foram canceladas, já que provavelmente sessões mais longas não ocuparam muito tempo o *buffer* de sessões ativas.

O gráfico da Figura 14 exibe a quantidade de requisições retiradas das filas. É importante observar que com a política de rejeição baseada em sessões ocorreram menos retiradas de requisições das filas, conseqüentemente menos perda de processamento, para todas as políticas utilizadas em conjunto com o CAS.

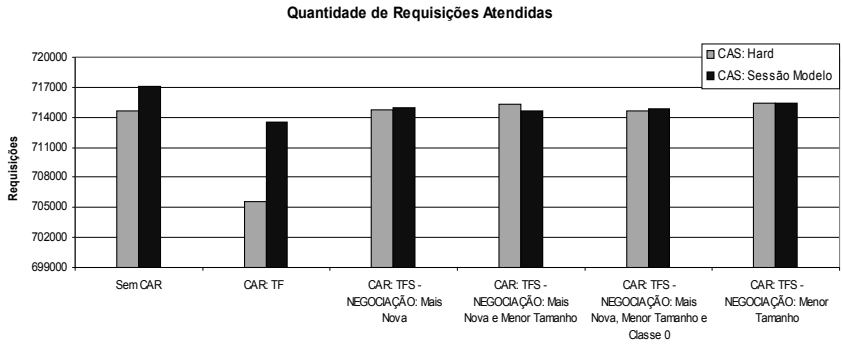


**Figura 13. Quantidade de requisições atendidas pelos mecanismos de CAR utilizados em conjunto com as políticas de CAS**

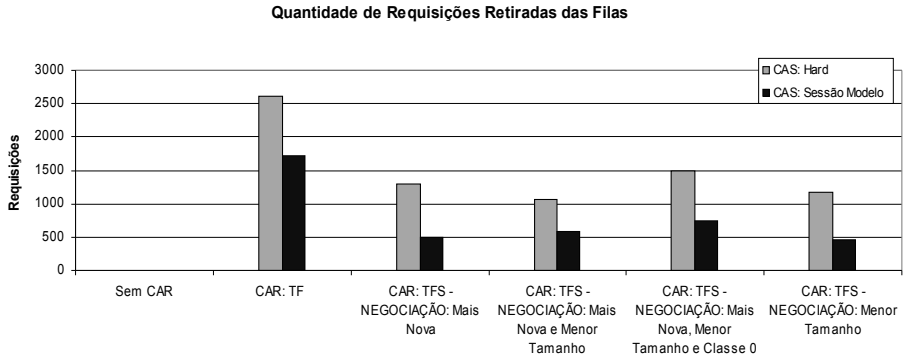


**Figura 14. Quantidade de requisições retiradas das filas pelos mecanismos de CAR utilizados em conjunto com políticas de CAS**

Para a política de negociação de descarte de sessões mais recentes e de menor tamanho, mais uma vez o CAS com BSM atende um maior número de sessões, sendo que para isso, menos requisições tiveram que ser retiradas das filas. Com isso, o algoritmo de negociação melhor se adequou a política BSM do CAS, para o qual cancelou menos requisições e conseqüentemente, menos sessões.

## 5.1. Considerações

Os resultados dos experimentos realizados nos mostram que ao se utilizar políticas de descarte de sessões para o CAS, maior quantidade de clientes ficam satisfeitos. No entanto, as sessões destes clientes são curtas e com menor quantidade de requisições emitidas. Essas características de sessões não se enquadram nas encontradas nos web sites de vendas, onde sessões mais longas prevalecem [Cherkasova and Phaal 1999].

Portanto, para melhor utilização das políticas desenvolvidas, deve-se conhecer previamente, as características da carga de trabalho que serão impostas aos servidores web.

# 6. Conclusões

A introdução de reconhecimento de sessões http em um modelo de servidor web com serviços diferenciados (SWDS) contribuiu para adaptar o modelo a uma das principais e crescentes demandas da Internet, aplicações web que utilizam sessões, como as encontradas em sites *e-business*.

O novo esquema para controle de admissão de sessões (CAS) foi desenvolvido e introduzido no modelo SWDS, considerando duas políticas para aceitar novas sessões, com garantia de finalização. A política Baseado em Sessão Modelo (BSM) estima a capacidade do sistema de aceitar novas sessões baseando-se em um modelo de sessão construído dinamicamente a partir da carga do sistema. Este mecanismo demonstrou melhor desempenho que o mecanismo da política *Hard-Threshold*, uma vez que manteve o sistema menos sobrecarregado com sua utilização.

O controle de admissão de requisições (CAR) utilizando o mecanismo de Admissão por Tamanho de Fila Com Negociação, manteve o sistema livre de sobrecargas e ofereceu atendimento diferenciado para as sessões. As políticas de negociação para descarte de sessões desenvolvidas tiveram um papel importante ao contribuir com a priorização do atendimento das sessões.

Enfim, os resultados obtidos mostram que os controles propostos constituem estruturas fundamentais para a estabilidade do desempenho do sistema, bem como os mecanismos desenvolvidos têm grande importância no atendimento das sessões e, portanto, de seus clientes. Contudo, através de uma abordagem baseada em diferenciação de atendimento foi possível incorporar ao modelo SWBS requisitos de QoS à sessões.

## Referências

Andreolini, M., Casalicchio, E., Colajanni, M., and Mambelli, M. (2004) “A Cluster-Based Web System Providing Differentiated and Guaranteed Services”, Cluster Computing 7, 1 (Jan. 2004), p. 7-19.

Arlitt, M. (2000) “Characterizing Web user sessions”, SIGMETRICS Perform. Eval. Rev. 28, 2 (Sep. 2000), p. 50-63.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998) “An Architecture for Differentiated Services”, RFC 2475, IETF.

Chen, H. and Mohapatra, P. (2003) “Overload control in QoS-aware web servers”, Comput. Networks 42, 1 (May. 2003), p. 119-133.

Cherkasova, L. and Phaal, P. (1999) “Session-Based Admission Control: A Mechanism for Improving Performance of Commercial Web Sites”, In Proceedings of IEEE/IFIP IWQoS’99. (31 May- 4 Jun. 1999).

Cherkasova, L. and Phaal, P. (2002) “Session-Based Admission Control: A Mechanism for Peak Load Management of Commercial Web Sites”, IEEE Trans. Comput. 51, 6 (Jun. 2002), p. 669-685.

Fishwick, P. A. (2004) “SimPackJ: Version 1.0”, University of Florida. (Nov. 2004), http://www.cise.ufl.edu/~fishwick/simpack

Teixeira, M. A. M., Santana, M. J. and Santana, R. H. C. (2005) “Servidor Web com Diferenciação de Serviços: Fornecendo QoS para os Serviços da Internet”, XXIII Simpósio Brasileiro de Redes de Computadores. Fortaleza, CE, 2005.

Traldi, O. A., Barbato, A. K., Santana R. H. C., Santana, M. J. and Teixeira, M. A. M. (2006) “Algoritmos de Diferenciação de Serviços para Servidores Web com Suporte à QoS”, XII Simpósio Brasileiro de Sistemas Multimídia e Web. Natal, RN, 2006.