Política de Escalonamento para Servidores Web baseada na Velocidade da Conexão

Cristina Duarte Murta[1], Tarcísio Paulo Corlassoli[2]
[1] Departamento de Informática Universidade Federal do Paraná
Curitiba - Paraná - Brasil
{cristina@inf.ufpr.br}
[2] Departamento de Informática Centro Federal de Educação Tecnológica do Paraná
Via do conhecimento, Pato Branco, Paraná, Brasil
{tpc@pb.cefetpr.br}

Resumo

Este artigo apresenta uma nova política de escalonamento para o processamento de requisições HTTP estáticas em servidores Web. Esta nova política chama-se FCF (*Fastest Connection First*). A política proposta atribui prioridades às requisições HTTP baseando-se no tamanho do arquivo solicitado e na velocidade da conexão com o usuário. As requisições para arquivos menores feitas através de conexões mais rápidas recebem maior prioridade. O que motivou a proposição desta política de escalonamento foi a variabilidade dos tamanhos dos arquivos transferidos na Web, a diversidade de condições de conectividade observadas na Internet e a possibilidade de saber com antecedência o tamanho do processo para atender a requisições estáticas. O objetivo da política FCF é otimizar a interação entre servidor Web e Internet visando um menor tempo final de resposta. Os resultados apresentam evidências de que as diferenças de conectividade observadas na Internet afetam o desempenho do servidor, e que essa informação pode ser utilizada para melhorar significativamente o desempenho do sistema.

Abstract

This paper shows a new scheduling policy for the processing of static HTTP requests in Web servers. This policy, called FCF (Fastest Connection First), gives priority to HTTP requests based on the size of the requested file and on the speed of the user's connection. The requests for smaller files through faster connections receive the highest priorities. The motivation of this proposal is the variability of the file sizes transferred in the Web, the diversity of the effective bandwidth of the user's connection observed in the Internet and the possibility of knowing the size of the process in advance. The results show evidences that the different levels of connectivity of the Internet users affect the performance of the Web server, and that this information can be used to improve the performance of the system.

1 Introdução

Servidores Web tratam requisições oriundas de um imenso e heterogêneo universo de usuários. A diversidade de usuários diz respeito não apenas à cultura, idioma e localização geográfica, mas também a plataformas de hardware, software e, principalmente, a condições de conectividade. Pesquisa realizada pelo *Graphics, Visualization, & Usability Center* do

Georgia Institute of Technology [08] mostrou que, em 1998, as condições de conectividade variavam até seiscentas vezes, por exemplo, de um modem de 14,4 Kbps para uma conexão de 10 Mbps. Além da largura de banda, fatores dinâmicos como a quantidade de tráfego no enlace, o número de *hops* no caminho de uma conexão TCP, o atraso, a taxa de transmissão nos enlaces intermediários e características das diversas implementações do protocolo TCP contribuem para a variabilidade das condições de conectividade observadas na Internet. Alguns trabalhos mostram que as variações na Internet exercem grande influência no desempenho dos servidores Web [05, 17, 18]. No entanto, apesar da observação desta variabilidade e de sua influência sobre o desempenho destes servidores, não são encontrados na literatura trabalhos sobre políticas de escalonamento para servidores Web que levem em consideração a interação do servidor com a Internet.

Além da variabilidade nas condições de conectividade, outras duas características do ambiente Web podem ser exploradas no projeto de novas políticas de escalonamento. Estas características são a possibilidade de conhecer antecipadamente o tamanho dos processos para tratamento de requisições HTTP estáticas e a variabilidade do tamanho dos arquivos requisitados. Para requisições HTTP estáticas, que são a maioria das requisições Web [03], o tempo de processamento pode ser considerado proporcional à quantidade de bytes servidos [04, 07, 12, 13]. Desta forma, é possível prever o tamanho do processo antes do atendimento da requisição. A grande variabilidade nos tamanhos dos arquivos da Web é amplamente reconhecida na literatura. Por exemplo, ver [11, 14].

O objetivo deste trabalho é propor uma política de escalonamento que reconheça e explore os benefícios das características apontadas: diversidade das condições de conectividade dos usuários na Internet; variabilidade dos tamanhos dos arquivos transferidos e possibilidade de conhecer com antecedência o tamanho dos processos. A política pode ser aplicada tanto em servidores Web como em Web caches, uma vez que ambos trabalham de maneira semelhante quanto ao tratamento de requisições estáticas. A idéia é explorar a variabilidade nas condições de conectividade dos usuários da Web. A proposta é dar prioridade para a requisição cuja conexão será encerrada primeiro, isto é, aquela que terá o menor tempo de duração. A conexão mais rápida é a que tem a menor requisição feita através da rede com as melhores condições de transmissão (maior banda, menor latência, menor congestionamento). A política proposta foi denominada *Fastest Connection First* (FCF). Intuitivamente, uma política de escalonamento que priorize requisições originárias de conexões mais rápidas pode reduzir o tempo de serviço destas requisições, com conseqüente diminuição do tempo de duração das respectivas conexões e redução do número de requisições pendentes. Requisições pendentes são aquelas cujas conexões já poderiam ter sido encerradas pelo servidor, mas que ainda não foram concluídas devido à demora na transferência pela Internet. A finalização de uma conexão indica que os recursos do servidor estarão disponíveis para o atendimento de novas requisições.

Além disso, garantir maior prioridade para usuários com melhor conectividade auxilia o servidor a manter um tempo médio de resposta condizente com as expectativas dos usuários. Aqueles que possuem conexões mais velozes esperam respostas rápidas. Se a conexão é rápida, a requisição também será tratada rapidamente no servidor. O ponto fundamental desta política é garantir que os *buffers* das conexões mais rápidas recebam mais rapidamente os dados do sistema de arquivos. Quando as requisições são atendidas por servidores sobrecarregados, o tempo de serviço pode ser perceptível por um usuário que possui este tipo de conexão. Para usuários com conexões mais lentas, o tempo devido à espera no servidor é muito menos perceptível.

A política proposta foi comparada com outras duas políticas de escalonamento, a política FIFO (*First In First Out*), padrão dos servidores Web, e a política SRPT (*Shortest Remaining Processing Time*). A avaliação foi feita através de simulação, na qual foram modelados três componentes do ambiente Web, a saber: os usuários, o servidor Web e a

Internet. Os usuários, que representam a carga de serviço imposta ao servidor, foram modelados a partir dos conceitos e fórmulas do gerador de cargas denominado SURGE [03, 14] (*Scalable URL Reference Generator*). Do servidor Web foram simulados os três principais recursos, CPU, disco e interface de rede. Para calcular o tempo de transmissão na Internet foi modelado o funcionamento dos protocolos TCP e HTTP, considerando as variações do RTT (*round trip time*), das condições de conectividade, incluindo congestionamentos e perdas de pacotes. Os resultados demonstraram que é possível reduzir o tempo médio de resposta do usuário ajustando a interação entre o servidor Web e a Internet.

Este artigo está organizado em seções. A próxima seção descreve a política FCF. A seção 3 apresenta o modelo de simulação elaborado para avaliação do desempenho da política, bem como a descrição do experimento realizado para comparação da política FCF com outras duas políticas de escalonamento. A seção 4 apresenta e discute os resultados do experimento realizado e a útlima seção apresenta as conclusões.

2 A Política Fastest Connection First (FCF)

2.1. Descrição da Política FCF

Um servidor Web geralmente atende várias requisições concorrentemente. Alguns servidores implementam esta multiprogramação criando um novo processo ou *thread* para cada nova conexão. Dependendo da implementação, o custo para criar um novo processo pode ser alto. Alguns servidores, na tentativa de minimizar este custo, implementam um mecanismo conhecido como *pool* de processos, onde um determinado número de processos é criado durante a inicialização do servidor [06]. Seja qual for o caso, o servidor não faz uso de prioridades para o tratamento das requisições [01, 07].

O servidor Web Apache, por exemplo, cria um *pool* de processos e trata as requisições de acordo com a ordem de chegada (FIFO), não importando seu tipo ou tamanho. As requisições admitidas no *pool* são atendidas, concorrentemente, pela política PS (*Processor Sharing*). Assim, a escolha do processo que pode utilizar um recurso do sistema (processador, disco ou interface de rede) é feita pela ordem de chegada da requisição.

A política FCF muda a ordem na qual as requisições ou processos recebem recursos do sistema. O próximo processo a receber recursos será aquele que estiver atendendo à requisição de maior prioridade. Esta prioridade será definida em função de dois parâmetros: menor quantidade de bytes a processar e maior taxa estimada de transmissão. A idéia é dar prioridade ao processo cuja conexão correspondente pode ser encerrada mais rapidamente, isto é, à menor requisição feita através da conexão com a maior taxa de transmissão. Assim, o primeiro parâmetro é utilizado para dar prioridade ao menor processo e o segundo parâmetro visa dar prioridade à requisição que pode ser transmitida no menor tempo possível. A taxa de transferência depende das condições de conectividade do usuário e das variações dinâmicas da Internet.

A política FCF opera da seguinte forma. Primeiramente busca-se na fila de requisições aquela com o menor número de bytes a processar. A seguir, outra pesquisa é realizada na mesma fila, buscando-se a requisição em processamento que tem a maior taxa de transmissão estimada. Esta nova pesquisa, porém, é restrita às requisições que possuírem um número de bytes a processar menor ou igual ao produto de uma constante (*beta*) maior que um, pelo número de bytes encontrado na primeira pesquisa. *Beta* define a abrangência da faixa de pesquisa. Ao variar o valor de *beta* damos maior ou menor peso para a taxa de transmissão ou para o tamanho do processo. O cálculo da taxa de transmissão é feito dividindo-se o tempo estimado para transferir o arquivo pelo tamanho do arquivo. Quanto maior for esta taxa, maior será a prioridade da requisição.

O objetivo da política é considerar as condições de conectividade do usuário e todos os aspectos inerentes à transmissão de dados pela Internet. Conexões mais velozes tendem a ter RTT menores e janelas de receptor e congestionamento maiores [09]. Consequentemente, as requisições feitas através destas conexões, se atendidas com maior prioridade no servidor, podem ser concluídas mais rapidamente, liberando os recursos do servidor para o atendimento de novas requisições.

Requisições Web são servidas através de uma conexão TCP. Neste protocolo, cada pacote ou conjunto de pacotes de dados, enviado pelo remetente, deve ser confirmado pelo destinatário. Devido ao controle de congestionamento realizado pelo TCP (s*low start*), apenas um número limitado de pacotes pode ser enviado a cada vez, devendo os demais aguardar confirmação da chegada dos pacotes já enviados. As confirmações das conexões velozes chegam mais rapidamente e, consequentemente, novos pacotes podem ser enviados. A política FCF visa garantir que os *buffers* de *socket* das conexões velozes sempre possuam dados para enviar quando as confirmações chegarem. Ou seja, os pacotes já estarão disponíveis para o envio quando puderem ser enviados. A política FCF procura tratar a requisição no servidor de acordo com o nível de conectividade medido dinamicamente para a respectiva conexão. Se a conexão é rápida, a requisição também será tratada rapidamente no servidor. Portanto, o ponto fundamental desta política é garantir que os *buffers* das conexões mais rápidas recebam mais rapidamente os dados do sistema de arquivos. Desta forma, quando um ACK de uma conexão rápida chega, não haverá necessidade de ler os dados do sistema de arquivos pois estes já estarão no *buffer*. Em situações em que o servidor Web opera com uma carga moderada ou baixa, quando uma confirmação chega, os *buffers* das conexões normalmente já têm dados a transmitir [16] independente da política de escalonamento aplicada. Contudo, quando o servidor opera com uma carga elevada, dar prioridade às conexões rápidas trará mais garantias de que seus *buffers* tenham dados a transferir quando da chegada de confirmações (ACK). Além disso, com carga elevada tornam-se maiores as filas de mensagens de controle da saída do TCP, de datagramas IP e de pacotes na interface de rede. Desta forma, a política de escalonamento poderia ser aplicada também nestas filas, de modo a garantir que os pacotes das conexões mais velozes cheguem mais rapidamente à interface de rede. O trabalho realizado em [13] aplicou prioridade baseada no tamanho do arquivo apenas na fila de pacotes da interface de rede, obtendo bons resultados.

Outra contribuição importante desta política de escalonamento é a de proporcionar redução do número de requisições pendentes. Como poderá ser observado nos resultados, esta redução acontece em decorrência do atendimento prioritário dado às requisições que são transmitidas mais rapidamente pela Internet. A redução do número de requisições pendentes proporciona uma conseqüente redução das trocas de contexto, as quais, como observado no trabalho [17], podem prejudicar o desempenho de servidores Web. A comprovação de que o excesso deste tipo de requisição prejudica o desempenho de servidores Web foi apresentada no trabalho [18]. Os resultados deste trabalho demonstraram que o servidor Web Apache, trabalhando sobre uma rede com RTT médio de 200 milissegundos, pode ter seu desempenho reduzido em mais de 50% comparativamente com o trabalho sobre uma rede local. A redução no desempenho foi provocada justamente pelo aumento das trocas de contexto e pela necessidade de mais memória para gerenciar um maior número de requisições concorrentes.

3 Modelo de Simulação para a Política FCF

A políticas FCF, FIFO e SRPT foram avaliadas e comparadas através de simulação. O experimento de simulação modelou, igualmente para as três políticas, os três principais componentes do ambiente Web, a saber: os usuários que representam a carga de serviço, o

servidor Web e a Internet. No decorrer desta seção será descrito como foram simulados estes componentes.

3.1 Simulação do Servidor

Um servidor Web é um programa que aceita conexões com o objetivo de servir requisições de acesso às páginas Web armazenadas no seu sistema de arquivos. Estas informações podem ser obtidas de forma estática ou dinâmica. O objetivo desta simulação é avaliar o servidor Web no processamento de requisições de conteúdo estático. O atendimento deste tipo de requisição utiliza majoritariamente três recursos do sistema: CPU, disco e interface de rede. O processamento das requisições estáticas no servidor pode ser descrito da seguinte forma: estabelecimento da conexão TCP, recebimento do pedido HTTP, processamento do cabeçalho da requisição, leitura do arquivo do disco ou do *cache*, empacotamento e envio dos dados para o usuário, fechamento da conexão e registro no arquivo de *log*. De acordo com os trabalhos [05, 07, 12], dentro desta seqüência de ações a demanda de serviço para uma requisição pode ser dividida em três partes, a saber:

- tempo de CPU: representa o tempo gasto pela CPU para fazer o *parser* da requisição, empacotamento e desempacotamento dos segmentos TCP e datagramas IP, controle da transferência dos dados do disco para o *buffer* de *socket* e deste para a interface de rede, entre outros aspectos da manipulação de uma requisição HTTP;
- tempo de leitura: é o tempo gasto pelo disco para transferir o arquivo solicitado para o *buffer* do *socket*;
- tempo de processamento na interface de rede: tempo gasto para transferir para o *buffer* da interface de rede os dados solicitados.

Seguindo esta divisão, cada uma das três partes da demanda de serviço de uma requisição pode ser associada a um dos três recursos do sistema, CPU, disco e interface de rede. Desta forma, o servidor Web foi simulado a partir de três filas, uma para cada recurso citado. Os trabalhos [09, 12, 13] também utilizam modelagem semelhante.

As entidades armazenadas nas filas correspondem a requisições HTTP. Por considerar que nos estágios de recebimento e *parser* da requisição ainda não estão disponíveis as informações para definir a prioridade, a fila de CPU é gerenciada pelo modelo FIFO e suas requisições são processadas de acordo com a política PS. Este modelo é considerado uma abstração da política padrão utilizada em servidores Web [09, 15].

O processamento da fila de disco consiste em transferir dados do arquivo requisitado, passando estes dados do sistema de arquivos para o *buffer* do *socket*. Os dados são lidos do disco em blocos de 16 Kbytes (*block mode*). Depois de ler um bloco de dados, o descritor da conexão é transferido para a fila de rede. A fila da interface de rede terá o custo de transferir os dados do *buffer* do *socket* para o *buffer* da interface de rede. Neste ponto os pacotes podem ser enviados pela rede. Uma abordagem semelhante a esta é utilizada no trabalho [16]. Quando a requisição tiver todos os seus dados transferidos para o *buffer* da interface de rede ela é retirada da fila. Contudo, caso ainda existam dados a transmitir, a requisição deve voltar para a fila de disco aguardando que este envie novos dados para o *buffer* do *socket*.

A requisição somente estará concluída quando tiver cumprido a soma dos custos das filas de CPU, disco e rede, considerando que uma requisição não pode ser processada simultaneamente nos três recursos do sistema (o processamento é concorrente).

3.2 Simulação da Carga

O segundo componente do ambiente de simulação é a carga de serviço imposta ao servidor Web. Esta carga corresponde ao conjunto de requisições HTTP recebidas durante um período de tempo. Para avaliar corretamente o desempenho de um servidor Web é necessário gerar com precisão a carga de serviço. Para a geração desta carga utilizamos o *benchmark* SURGE (*Scalable URL Reference Generator*) [03, 14]. O SURGE reproduz as principais características da carga Web e é configurável. Todos os conceitos e fórmulas deste gerador foram utilizados na avaliação das políticas FCF, FIFO e SRPT. O SURGE gera um fluxo contínuo de requisições que simula o acesso de centenas ou milhares de usuários. Estes acessos reproduzem as principais características de uma carga real de serviço imposta a um servidor Web. Dentre estas características estão a distribuição de cauda pesada dos tamanhos de arquivo, a localidade temporal, a popularidade dos arquivos, o número de arquivos por página, o *think time* e o número de usuários simultâneos. Embora tenham sido modeladas todas as características citadas, apenas o número de usuários simultâneos será descrito com detalhes por ser de maior importância para o entendimento da simulação.

Geradores de carga, como o SURGE, utilizam o conceito de usuários equivalentes para representar uma população de usuários acessando um servidor. No SURGE, um usuário equivalente (UE) é um processo que faz requisições ao servidor e aguarda as respostas. Entre cada requisição efetuada pelo UE existe um tempo de inatividade que corresponde ao *think time*. Este modelo de geração de carga apresenta rajadas de requisições seguidas de longos tempos de inatividade (*think time*). Estas rajadas representam o fato das páginas Web geralmente incluirem referências a dezenas de outros objetos, gerando novas requisições HTTP. Depois de um período de inatividade (*think time*) o usuário requisita uma ou mais páginas e o próprio *browser* se responsabiliza por requisitar, em um pequeno intervalo de tempo, os objetos referenciados. A intensidade da carga gerada pelo SURGE é definida pelo número de UEs. Nesta simulação o número de UEs foi variado de 70 a 700.

3.3 Simulação da Internet

Fazer a modelagem de toda a topologia da Internet é um problema que ainda não foi solucionado. As dificuldades para tal modelagem e simulação são apresentadas em [21], dentre elas citamos tamanho imenso, heterogeneidade técnica e administrativa e elevada taxa de crescimento. Contudo, o estudo e modelagem de alguns aspectos da Internet são realizados por inúmeros trabalhos, dentre os quais destacamos [02, 09, 10, 19, 20]. Embora nenhum destes trabalhos tenha realizado uma análise completa de toda a Internet, eles permitiram que certos aspectos da grande rede tenham sido decompostos e modelados.

Neste trabalho a Internet foi simulada modelando-se o funcionamento dos protocolos TCP e HTTP/1.1. Para isto, utilizamos os seguintes parâmetros: RTT médio, largura de banda, quantidade de tráfego disputando os enlaces e perda de pacotes devido a congestionamentos. A seguir serão apresentados valores obtidos em alguns trabalhos, os quais foram utilizados como base para a distribuição do RTT e da largura de banda. A partir de um tamanho de arquivo, de um RTT médio, de um valor para a largura de banda e da modelagem dos protocolos é calculado o tempo total de transmissão do arquivo.

O RTT é modelado a partir do RTT mínimo. Este, por sua vez, é o menor tempo possível, gasto por um único pacote ou datagrama, para ser enviado de um *host*, chegar à outro *host* e retornar. O RTT é determinado pela velocidade da luz, pelo retardo imposto pelos pontos de conexão (roteadores, protocolos, etc.) e pelo atraso devido ao tráfego da rede. Desta forma, o RTT varia de acordo com a carga à qual a rede está sendo submetida. Quanto maior o tráfego na rede, maiores serão as filas nos roteadores e maior será a latência. Em geral, as medidas de RTT possuem desvio padrão alto. No trabalho [09] foi apresentado um

estudo do RTT sobre 90 enlaces que interligam *hosts* que foram agrupados em três classes, a saber: comerciais, acadêmicos e estrangeiros. Este estudo foi realizado nos Estados Unidos. Os enlaces para *hosts* comerciais e acadêmicos apresentaram os menores RTTs mínimos por estarem fisicamente mais próximos, a maioria dentro dos Estados Unidos e alguns no Canadá. Os enlaces para *hosts* estrangeiros tiveram os maiores RTTs mínimos justamente por estarem mais afastados. Foram avaliados *hosts* de quatorze países. Enlaces para *hosts* comerciais próximos tiveram um RTT mínimo que variou de 10 a 20 ms, os mais distantes tiveram um RTT mínimo entre 70 e 90 ms. Os enlaces para *hosts* acadêmicos tiveram um RTT mínimo na faixa de 20 a 120 ms e os enlaces para *hosts* estrangeiros entre 90 e 600 ms.

Com base nas informações do artigo [09] foram estabelecidas, para este trabalho, quatro classes de RTT mínimo, a saber: classe **X,** 20 ms; classe **Y,** 50 ms; classe **Z,** 90 ms e classe **W,** 280 ms. Foi considerado que 25% dos RTTs são da classe **X**, 35% da classe **Y,** 15% da classe **Z** e 25% da classe **W**. As classes **X** e **Y** têm por objetivo simular acessos de usuários próximos e medianamente próximos. As classes **Z** e **W** visam simular acessos de usuários geograficamente mais distantes.

Uma simulação mais realística do RTT envolve não apenas o RTT mínimo mas também as variações decorrentes do tráfego na rede, que pode apresentar situações de congestionamento. As condições de tráfego geram variações no RTT que podem ser simuladas por distribuição de cauda pesada como a de Pareto [09]. Para simular esta variabilidade, o RTT foi modelado por uma distribuição de Pareto limitada, que tem como parâmetro o RTT mínimo de cada classe. O RTT gerado foi limitado a oito vezes o RTT mínimo. Outro parâmetro exigido pela distribuição de Pareto, além do valor mínimo, é o grau de variabilidade, denominado alfa, que foi fixado em 1,4. Este modelo de simulação dos RTTs gerou medidas semelhantes às apresentadas no artigo [02], no qual, para 5262 *hosts* servidores, observou-se uma média de 241 ms e um desvio padrão de 435 ms para o RTT. A modelagem da Internet a partir destes valores gerou resultados coerentes também com o trabalho [23], onde foi demonstrado que o tempo médio de duração de requisições HTTP na Internet fica entre 2 e 4 segundos. Estes resultados contribuem para validar a simulação.

As velocidades de transmissão (larguras de banda) definidas para a simulação e sua relação com o RTT são apresentadas na Tabela 1. As freqüências para as velocidades de transmissão foram obtidas de uma pesquisa realizada pelo *Graphics, Visualization, & Usability Center* do *Georgia Institute of Technology* com 2710 usuários de diversos países em outubro de 1998 [08]. É importante observar que a característica que motiva a proposta da política FCF é a variabilidade das condições de conectividade dos usuários. Acreditamos que dados mais recentes possam alterar a freqüência apresentada na Tabela 1 mas não acreditamos que estes dados impliquem numa diminuição da variabilidade das condições de conectividade. Pelo contrário, é provável que tenha havido um aumento da variabilidade, uma vez que enquanto já há redes de alta velocidade em operação, nada indica que houve progressos para os usuários com conexões lentas. Na simulação realizada, cada usuário é enquadrado em uma linha da Tabela 1, com as características definidas. As velocidades iguais ou acima de 128 Kbps são enquadradas somente nas classes X e Y de RTT. As velocidades abaixo de 128 Kbps somente nas classes Z e W.

Velocidade	Freqüência (%)	Classe de RTT
14 Kbps	2	90 e 280
28 Kbps	4	90 e 280
33 Kbps	12	90 e 280
56 Kbps	34	90 e 280
128 Kbps	19	20 e 50
1 Mbps	13	20 e 50
4 Mbps	9	20 e 50
10 Mbps	7	20 e 50

Tabela 1 : Velocidade de conexão do usuário e RTT associado.

4 Resultados

A principal contribuição deste trabalho é a proposta e a avaliação da política FCF e sua comparação com as políticas FIFO e SRPT. A política FIFO foi escolhida por ser padrão em servidores Web. SRPT foi escolhida porque utiliza o critério de tamanho de tarefa, que também é adotado pela política FCF. SRPT tem um desempenho comprovadamente superior à FIFO. Nosso objetivo ao utilizá-la na comparação de resultados é verificar se há ganho de desempenho quando são aplicados adicionalmente critérios de escalonamento baseados nas condições de conectividade.

Todos os experimentos foram realizados com três conjuntos de políticas para comparação. O primeiro conjunto, denominado FIFO nos gráficos deste trabalho, consiste na implementação da política FIFO para as filas de disco e de rede. O segundo conjunto, denominado FCF, implementa a política FCF nas filas de disco e de rede. O terceiro conjunto, chamado de SRPT, implementa a política SRPT nas filas de disco e rede. Para todos os conjuntos, as requisições são admitidas na CPU pela política FIFO e processadas por PS.

A carga de serviço utilizada na simulação foi gerada tendo como base um servidor Web com 10.000 arquivos diferentes. O arquivo mais acessado teve 102.500 acessos. Em cada execução da simulação foram feitas 1.006.149 requisições ao servidor sendo que, deste total, 545.876 foram requisições diretas de usuários (arquivo base) e as restantes foram de requisições a objetos referenciados para compor uma página (imagem, som, vídeo, entre outros). O tamanho médio dos arquivos foi de 14.072 bytes. O número de UEs foi variado de 70 a 700. Sob o ponto de vista do servidor, cada uma das políticas foi avaliada com a mesma carga de trabalho, isto é, os mesmos arquivos foram requisitados na mesma seqüência. Sob o ponto de vista do usuário, alguns usuários podem fazer mais ou menos requisições de acordo com a prioridade dada a eles pela política de escalonamento. Isto ocorre porque, no SURGE, o usuário que recebe resposta primeiro faz a próxima requisição da carga de serviço modelada para o servidor. Desta forma, se algum usuário tem maior prioridade, recebe respostas antes que os demais e, consequentemente, faz mais requisições.

A constante *beta* teve valor igual a dois na simulação. Desta forma, a partir da definição do menor arquivo da fila, um arquivo com até o dobro do tamanho poderá ter maior prioridade se for requisitado por um usuário com conexão mais rápida.

É importante ressaltar que muitos resultados das políticas FCF e SRPT são semelhantes pelo fato de serem políticas que compartilham o mesmo critério de escalonamento. Ambas visam priorizar as requisições menores. A diferença é que FCF considera também aspectos da Internet para realizar o escalonamento conseguindo, desta forma, melhores resultados em métricas que dependem da Internet como, por exemplo, requisições pendentes e tempo de resposta. O objetivo da FCF é otimizar a interação entre o servidor Web e a Internet. A política SRPT, por ser a mais beneficiada pelas características da

carga de serviço e por não considerar a Internet, tem melhores resultados em métricas relacionadas diretamente ao servidor, como tempo de resposta no servidor. Contudo, deve ser observado que ambas as políticas (FCF e SRPT) apresentam resultados muito melhores que a política padrão de servidores Web (FIFO). O fato é que a política FCF, além de ter como objetivo obter melhores valores para métricas convencionais, visa também otimizar a interação entre o servidor e a Internet. Por exemplo, ela evita que o servidor empregue recursos desnecessários em conexões lentas, procurando manter o tempo de resposta (atraso) no servidor compatível com o tratamento recebido na Internet.

O ajuste dos parâmetros tamanho da tarefa e velocidade de conexão na política FCF é bastante delicado. Na política FCF, o que indica maior ou menor peso das condições de conectividade é a constante *beta*. Quanto menor for esta constante, mais próxima estará a política FCF da política SRPT. Portanto, é possível aumentar a diferença entre as políticas aumentando o valor de *beta*. Deve haver um compromisso entre os parâmetros tamanho da tarefa e velocidade da conexão. É importante utilizar um valor de *beta* que mantenha o ganho de desempenho obtido pela prioridade dada às requisições menores e que também consiga vantagem com a prioridade dada às conexões mais rápidas. Com o valor utilizado para *beta* nesta simulação, a política FCF apresentou o melhor resultado na principal métrica para avaliação de desempenho de um servidor Web – o tempo de resposta observado pelo usuário.

4.1 Caracterização da Carga de Serviço

Para facilitar a avaliação de uma possível situação de *starvation* dos processos grandes, a carga de serviço foi classificada pelo tamanho dos arquivos em oito classes. O gráfico da Figura 1 mostra o percentual de requisições e o percentual de bytes em cada classe de arquivo. Podemos observar que a grande maioria das requisições são para pequenos arquivos. Mais de 90% das requisições são para arquivos menores que 32 Kbytes e, por outro lado, apenas 1,47% das requisições são para arquivos maiores que 128 Kbytes. Inversamente proporcional ao percentual de requisições, podemos ver neste gráfico que as requisições para arquivos acima de 128 Kbytes representam quase 40% da carga imposta ao servidor, uma vez que a demanda de serviço é proporcional ao número de bytes. Esta propriedade da carga Web permite que seja utilizada uma política como SRPT sem que ocorra *starvation* dos processos grandes. Comprovações deste fato serão apresentadas no decorrer desta seção.

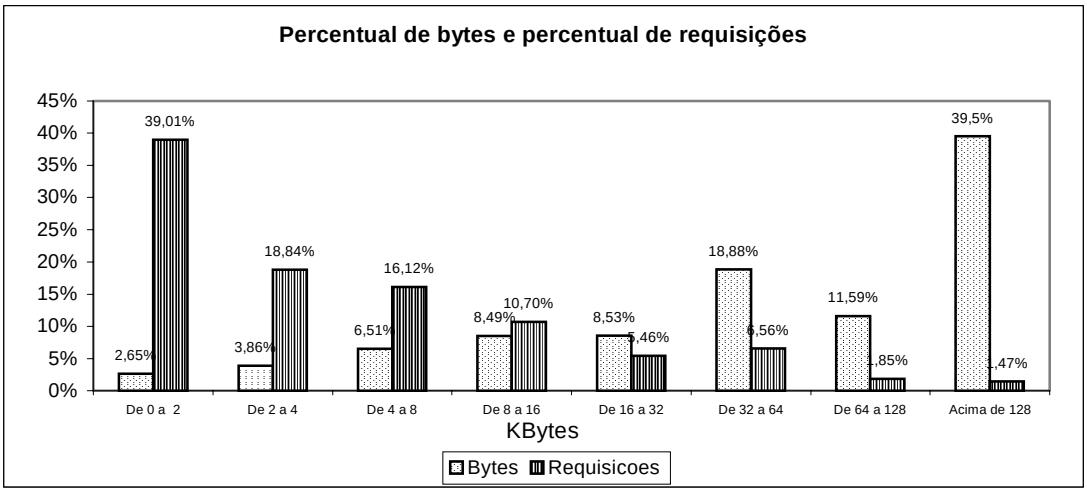

Figura 1 : Percentual de requisições por tamanho de arquivo.

4.2 Avaliação da Starvation Sofrida por Arquivos Grandes

Como podemos observar no gráfico da Figura 1, a classe dos arquivos acima de 128 Kbytes representa apenas 1,47% das requisições, mas corresponde a 39,50% dos bytes a serem processados e transmitidos. Portanto, essa classe, apesar de ter um número reduzido de requisições, representa grande parte da carga imposta ao servidor e à Internet. O fato de um pequeno número de requisições corresponder a grande parte da carga de serviço impede que processos grandes sejam demasiadamente prejudicados. Observamos no gráfico da Figura 1, que cerca de 85% dos arquivos requisitados são menores que 16 Kbytes e representam apenas 21,50% da carga. Dessa forma, mesmo dando-se prioridade máxima a 85% das requisições (pequenos arquivos), o servidor ainda vai dispor de 78,50% dos recursos para tratar os 15% restantes das requisições. Considerando-se esta característica da carga Web, não há necessidade da utilização de prioridades dinâmicas, como a utilizada no artigo [07], para evitar que ocorra *starvation* dos processos grandes. Essa mesma linha de raciocínio é apresentada nos trabalhos [11, 12, 13].

Para comprovar a não ocorrência de *starvation* apresentamos o gráfico da Figura 2, onde podemos observar o tempo médio de resposta visto pelo usuário para 700 UEs. Este gráfico é apresentado em escala logarítmica e mostra que apenas as requisições para arquivos com mais de 128 Kbytes sofrem pequena penalização nas políticas FCF e SRPT em relação à FIFO. Deve-se observar que esta classe de tamanho de arquivo é responsável por apenas 1,47% das requisições.

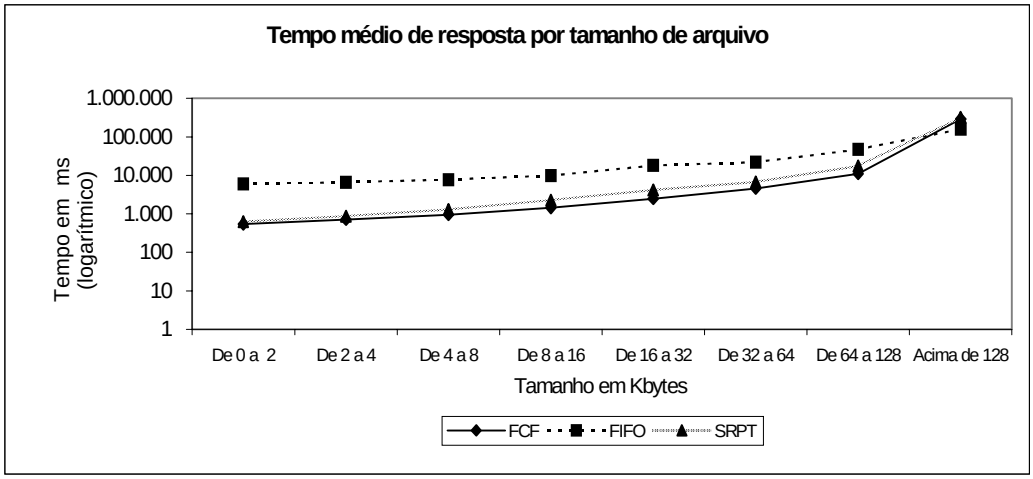

Figura 2 : Tempo médio de resposta por tamanho de arquivo para 700 UEs.

4.3 Avaliação da Starvation Sofrida por Requisições de Conexões Lentas

O gráfico da Figura 3 mostra o tempo médio de resposta visto pelo usuário em função da velocidade de conexão para 700 UEs. Observamos que os tempos acompanham as condições de conectividade, isto é, usuários de conexões mais rápidas têm melhores tempos de resposta para todas as políticas, mas esta observação é muito mais marcante para FCF. Deve ser ressaltado, contudo, que em nenhum caso a política FCF apresenta tempo médio de resposta superior à FIFO. Isso mostra que, devido às características da carga, é pouco provável a ocorrência de *starvation* nas requisições de conexões lentas. As conexões mais lentas apresentam um tempo médio de resposta superior na política FCF em relação à SRPT. Todavia, o propósito inicial da política FCF é dar maior prioridade às conexões mais rápidas

em detrimento às conexões mais lentas. FCF apresenta desempenho nitidamente superior para requisições de conexões rápidas.

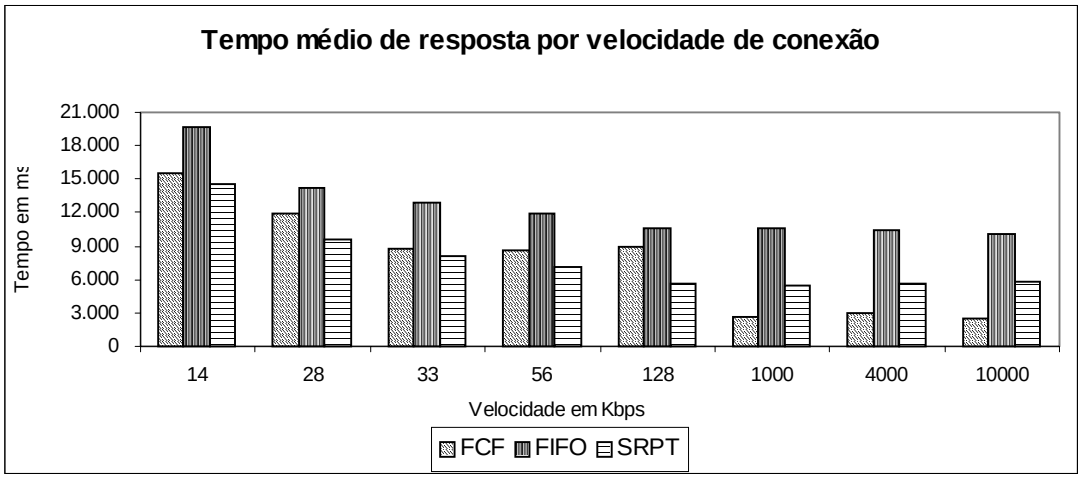

Figura 3 : Tempo médio de resposta por velocidade de conexão para 700 UEs.

4.4 Slowdown

Uma forma de avaliar a justiça de uma política de escalonamento é através da métrica chamada *slowdown*. O *slowdown* é calculado dividindo-se o tempo nas filas pela demanda de serviço nos três recursos do sistema (CPU, disco e rede). Um *slowdown* baixo significa que o servidor está tratando as requisições de forma compatível com seu tamanho.

O *slowdown* avalia a justiça das políticas em relação ao tempo de espera no servidor. O objetivo é manter este valor o mais baixo possível, garantindo boa relação entre o custo da requisição e a espera nas filas. O gráfico da Figura 4 apresenta os valores desta métrica.

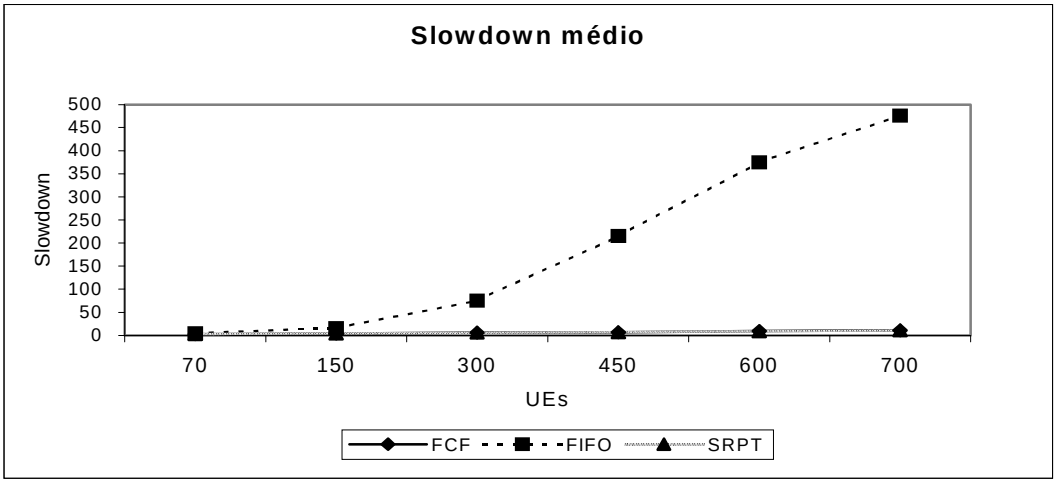

Figura 4: *Slowdown* médio.

Podemos observar que as políticas FCF e SRPT proporcionam maior justiça no tratamento das requisições. No pior caso, com 700 UEs, o *slowdown* da FIFO é cerca de quarenta vezes maior que o da FCF. A obtenção de um *slowdown* médio bastante baixo é decorrência, principalmente, do maior peso que a política de escalonamento atribui ao tamanho da requisição. Dessa forma, evita-se que pequenas requisições permaneçam muito tempo no servidor. No caso da FIFO, o atendimento de uma requisição muito grande obriga

as requisições comparativamente muito pequenas (que são a maioria), a aguardar na fila um tempo dezenas ou centenas de vezes maior que seu tamanho, o que contribui para aumentar o *slowdown*. Deve-se ressaltar que os valores do *slowdown* para as políticas SRPT e FCF são similares.

4.5 Requisições Pendentes no Servidor

Uma vantagem observada na política FCF, decorrente da prioridade dada às requisições de conexões velozes, refere-se ao número médio de requisições pendentes no servidor. Requisições pendentes são aquelas cujas conexões já poderiam ter sido fechadas pelo servidor, mas que ainda não foram concluídas devido ao atraso na transferência pela Internet. Para requisições de conexões lentas, mesmo que o servidor disponha de recursos para concluir seu processamento, isso não ocorre pois o envio de segmentos e recebimento de ACKs se prolonga por um tempo maior, obrigando o servidor a manter a conexão aberta. Por outro lado, priorizando-se conexões mais velozes, permite-se que as requisições que são transmitidas de forma mais rápida também sejam atendidas mais rapidamente pelo servidor. Em conseqüência, temos um menor número de requisições pendentes no servidor, como mostra o gráfico da Figura 5. A política FIFO apresenta um número médio de requisições pendentes cerca de 30% superior à política FCF. Neste gráfico também podemos constatar que a política SRPT, por não considerar os aspectos da Internet, mantém um número de requisições pendentes muito próximo ao da política FIFO.

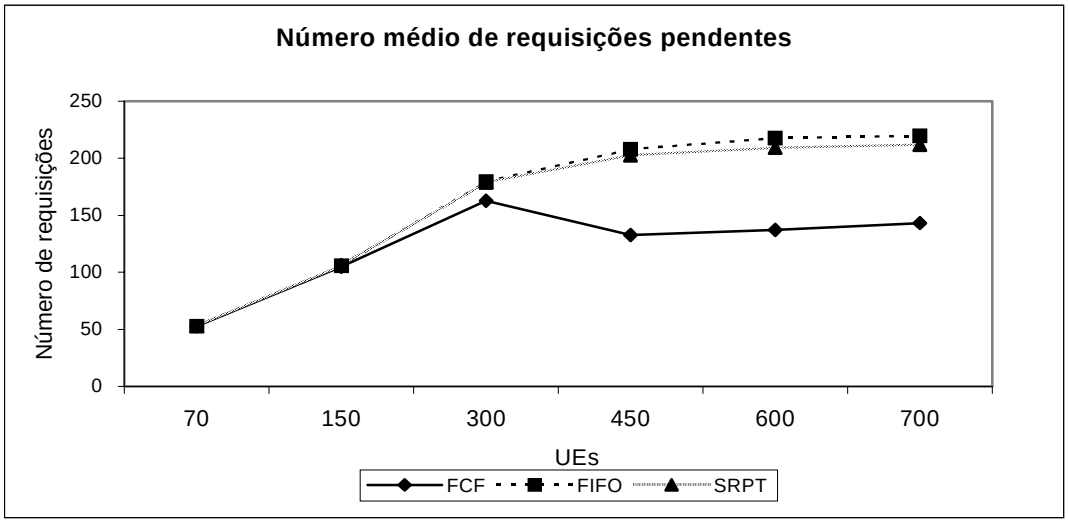

Figura 5 : Número médio de requisições pendentes.

A redução do número de requisições pendentes contribui para reduzir o número de trocas de contexto, o tamanho das estruturas de controle e a quantidade de memória necessária. Porém estas métricas não puderam ser avaliadas no ambiente de simulação utilizado. A avaliação destas métricas somente seria possível mediante a implementação real da política de escalonamento.

A comprovação de que o excesso de requisições pendentes prejudica o desempenho de servidores Web foi apresentada no trabalho [18]. Segundo os autores daquele trabalho, a redução no desempenho foi provocada justamente pelo aumento das trocas de contexto e pela necessidade de mais memória para gerenciar um maior número de requisições pendentes. O estudo apresentado no trabalho [05] também comprovou este fato. Os autores explicam que o problema das requisições que apresentam grandes latências de rede é que o servidor precisa

manter um processo ocupado com a requisição até algum tempo depois do envio do último segmento de dados ao usuário. Isto reduz o desempenho do servidor. Fato semelhante foi apresentado no trabalho [17].

4.6 Tempo Médio de Resposta no Servidor e no Cliente

As métricas mais importantes para a avaliação da nova política de escalonamento são o tempo médio de resposta no servidor e o tempo médio de resposta visto pelo cliente. O gráfico da Figura 6 mostra o tempo médio de resposta no servidor. Os valores apresentados correspondem à demanda de serviço nos três recursos do sistema (CPU, disco e rede) mais o tempo gasto nas filas dos respectivos recursos. Como podemos observar, os valores da opção FIFO são superiores aos da FCF que, por sua vez, são ligeiramente superiores ao da SRPT. Em média, esta métrica para a opção FIFO é 76% superior à FCF, e 95% superior à SRPT. O que garante a redução no tempo médio de resposta no servidor é o atendimento prioritário das requisições menores. Como este tipo de requisição corresponde à maioria das requisições e representa pequena fração da carga de serviço, reduzir seu tempo de resposta tem uma influência muito grande no tempo médio de resposta no servidor.

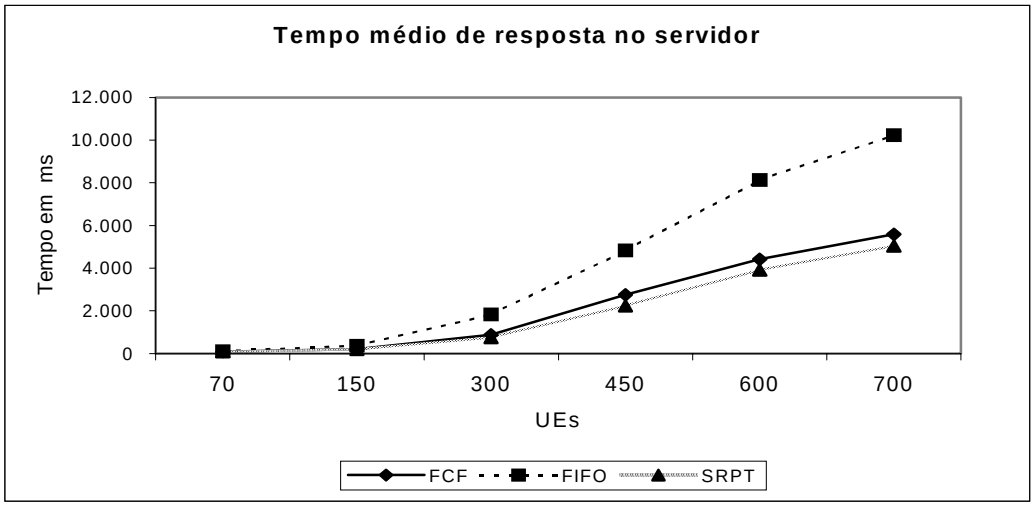

Figura 6 : Tempo médio de resposta no servidor.

Com relação à política FCF, cuja prioridade é estabelecida também em função da taxa de transmissão, o que observamos é uma pequena queda de desempenho, comparando-a à política SRPT. Contudo, esta queda de desempenho não é repassada ao tempo médio de resposta visto pelo usuário. Este fato pode ser constatado no gráfico da Figura 7. Este gráfico apresenta o tempo médio de resposta observado pelo usuário. Os valores apresentados incluem o tempo de resposta no servidor mais o tempo de transferência na Internet. Não é computado o tempo gasto pela máquina do usuário para exibir os dados transmitidos.

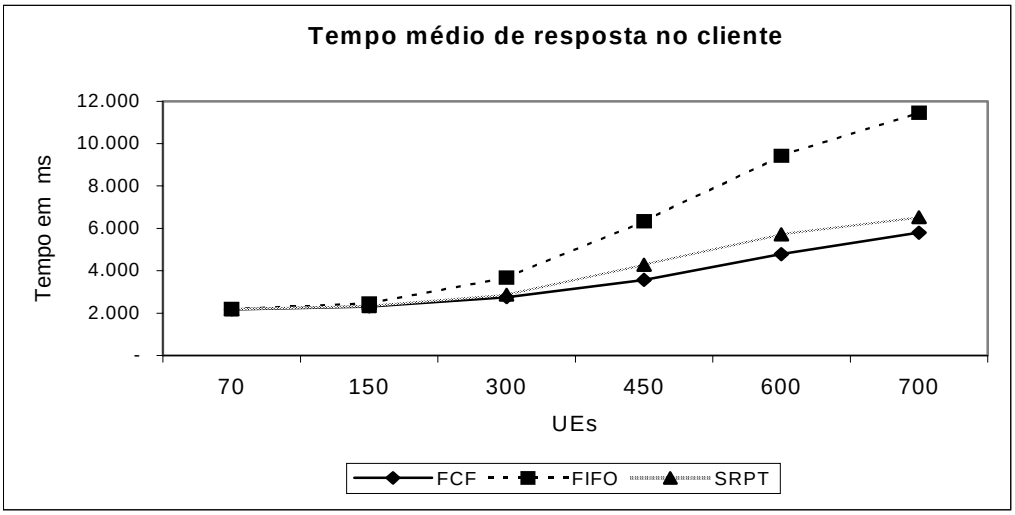

Figura 7 : Tempo médio de resposta para o usuário.

Neste gráfico observamos que a política FIFO novamente mostra o pior desempenho e que SRPT apresenta um desempenho intermediário. Por outro lado, o melhor desempenho é apresentado pela política FCF. As políticas FIFO e SRPT apresentam um tempo médio de resposta visto pelo usuário, respectivamente, 53% e 10% superior à FCF. Dois fatores contribuem para isso. Primeiro, a política FCF ajusta o tempo de resposta no servidor de acordo com o tempo de transmissão na Internet, ou seja, requisições que podem ser transmitidas com maior rapidez recebem mais recursos no servidor. Isso ajuda a explicar porque o tempo médio de resposta no servidor é menor para a política SRPT e o mesmo não ocorre com o tempo médio de resposta visto pelo usuário. O fato é que não adianta tratar com maior rapidez no servidor requisições que não terão um tratamento semelhante na Internet. Priorizar requisições de conexões lentas significa aumentar o número de requisições pendentes, como observado no gráfico da Figura 5. Dar prioridade às requisições de conexões mais velozes contribui diretamente para melhorar o tempo de resposta visto pelo usuário.

O segundo fator que explica o melhor tempo médio de resposta visto pelo usuário na política FCF é o aumento do número de requisições de usuários com conexões velozes. Uma vez que ocorre a redução do tempo médio de resposta para este tipo de usuário, a tendência é que eles façam um maior número de requisições comparativamente aos usuários de conexões lentas. Embora o aumento do número de requisições de usuários com conexões velozes não seja expressivo, ele contribui para a redução no tempo médio de resposta visto pelo usuário.

5. Conclusões

A necessidade de interação com a Internet e as características da carga de serviço tornam o servidor Web um ambiente computacional ímpar. A variabilidade das condições de conectividade observadas na Internet, aliada às características dos protocolos TCP e HTTP, afetam o desempenho do servidor. Considerar as características da carga de serviço e das condições de conectividade diminui o tempo de resposta às requisições, conforme foi mostrado neste trabalho através da política FCF (*Fastest Connection First*). Nestas condições, a política proposta demonstrou ser uma opção interessante para garantir um melhor desempenho de servidores Web.

O objetivo principal da política FCF foi melhorar o tempo de resposta para usuários Web através da otimização da interação entre o servidor Web e a Internet. Esta tentativa de otimização foi buscada atribuindo-se mais recursos no servidor às requisições que possuem mais recursos também na Internet, ou seja, podem ser transferidas mais rapidamente pela

rede. É correto afirmar que o funcionamento do protocolo TCP implicitamente atribui mais prioridade às requisições de conexões cujas mensagens de confirmação (ACKs) cheguem mais rapidamente no servidor. Contudo, esta prioridade dada pelo TCP ocorre apenas na fila de mensagens de controle do TCP. As demais filas do sistema não atribuem prioridade nenhuma às requisições de conexões mais rápidas. Além disso, uma vez que a requisição possui uma mensagem na fila do TCP, nada garante que seu processamento será prioritário se o tamanho desta fila for grande. Deste modo, a política FCF visa garantir, junto com o TCP, que as conexões rápidas recebam maior prioridade em todas as filas do servidor.

Com a política FCF observamos uma redução do tempo médio de resposta, sem penalização excessiva das requisições grandes e das requisições feitas através de conexões lentas. Ficou comprovado que o benefício dado às requisições de conexões velozes e às requisições menores não implica em demora excessiva para os demais usuários e requisições. Observamos através do *slowdown* que o tempo de resposta torna-se mais justo comparativamente ao tamanho da requisição. Priorizando-se conexões mais rápidas no servidor, estamos dando às requisições um tratamento compatível com o observado na Internet. Os usuários passam a esperar tempos compatíveis com suas condições de conectividade e os recursos do servidor não são desperdiçados com requisições de conexões pouco eficientes na Internet. Reduções significativas foram observadas no número de requisições pendentes, o que abre espaço para novos estudos sobre a influência que estas reduções trazem no desempenho do servidor.

Um fator que dificulta uma melhor avaliação da política FCF é que o modelo de simulação adotado, embora seja baseado em diversos trabalhos [07, 12, 13, 16], não é suficientemente detalhado para simular todos os aspectos da interação entre o servidor e a Internet. Não foram modeladas as filas dos protocolos TCP e IP. Além disso, as requisições são processadas no disco e na rede em blocos de 16 Kbytes. Desta forma, devido às características da carga Web, quase 85% das requisições são processadas em um único ciclo, distorcendo um pouco a avaliação da interação entre servidor e Internet. Para simular a interação exata do servidor com a Internet, além das filas já modeladas, seria necessário modelar também as filas dos protocolos TCP e IP. Seria necessário realizar o processamento individual de cada segmento e datagrama. Apesar destes aspectos, este trabalho apresenta evidências de que priorizar requisições pela velocidade de conexão pode melhorar o tempo de resposta dos usuários Web. Acreditamos que uma avaliação precisa da política somente possa ser realizada mediante uma implementação real, que é uma extensão óbvia do trabalho.

Referências Bibliográficas

[01] ALMEIDA, J.; DABU, M.; MANIKUTTY, A.; CAO, P. **Providing Differentiated Levels of Service in Web Content Hosting.** In Proceedings of Workshop on Internet Server Performance, Madison, Wisconsin, June 1998.

[02] CROVELLA, M. E.; CARTER, R. **Dynamic Server Selection in the Internet**. In Proceedings of HPCS'95: Third IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems, August 1995.

[03] BARFORD, P.; CROVELLA, M. **Generating Representative Web Workloads for Network and Server Performance Evaluation**. In Proceedings of ACM SIGMETRICS'98, Madison, July 1998.

[04] ALMEIDA, V. A. F.; ALMEIDA, J; YATES, D. **WebMonitor: a Tool for Measuring World-Wide Web Server Performance**. In Proceedings of Seventh IFIP Conference on High Performance Networking, White Plains, NY, April 1997.

[05] DILLEY, J.; FRIEDRICH, R.; JIN, T.; ROLIA, J. **Measurement Tools and Modeling Techniques for Evaluating Web Server Performance**. In Proceedings of 9th Int. Conf. on Modeling Techniques and Tools, vol. 1245 of Lecture Notes in Computer Science, p. 155-168. Springer-Verlag, 1997.

[06] MURTA, C. D.; ALMEIDA, J.; ALMEIDA, V. A. F. **Performance Analysis of a WWW Server**. In Proceedings of 22nd International Conference on Technology Management and Performance Evaluation of Enterprise-Wide Information Systems, San Diego, California, December 8-13, 1996.

[07] CHERKASOVA, L. **Scheduling Strategy to Improve Response Time for Web Applications**. In Proceedings of High-performance Computing and Networking: International Conference and Exhibition, p. 305-314, 1998.

[08] Tenth WWW User Survey (Conducted October 1998), Graphics, Visualization & Usability (GVU) Center at Georgia Tech. http://www.gvu.gatech.edu/user_surveys.

[09] ACHARYA, A.; SALTZ, J. **A Study of Internet Round-Trip Delay**. Technical Report CS-TR-3736, Department of Computer Science, University of Maryland, USA, December 1996.

[10] CARDWELL, N.; SAVAGE, S.; ANDERSON, T. **Modeling the Performance of Short TCP Connections.** Technical Report, Department of Computer Science and Engineering, Univ. of Washington, November 1998.

[11] HARCHOL-BALTER, M.; CROVELLA, M.; PARK, S. **The Case for SRPT Scheduling in Web Servers**. Technical Report MIT-LCS-TR-767, MIT Lab for Computer Science, October 1998.

[12] CROVELLA, M.; FRANGIOSO, B.; HARCHOL-BALTER, M. **Connection Scheduling in Web Servers**. In Proceedings of USITS '99: USENIX Symposium on Internet Technologies and Systems, p. 243-254, Boulder, Colorado, October 1999.

[13] HARCHOL-BALTER, M.; BANSAL, N.; SCHROEDER, B.; AGRAWAL, M. **Implementation of SRPT Scheduling in Web Servers.** Technical Report number CMU-CS-00-170. Carnegie Mellon School of Computer Science, October 2000.

[14] BARFORD, P.; CROVELLA, M. **An Architecture for a WWW Workload Generator.** In Wide Web Consortium Workshop on Workload Characterization, October 1997.

[15] BESTAVROS, A.; KATAGAI, N.; LONDOÑO, J. **Admission Control and Scheduling for High-Performance WWW Servers.** Technical report BUCS-TR-97-015, Boston University, Computer Science Department, August 1997.

[16] DRUSCHEL, P.; BANGA, G. **Lazy Receiver Processing (LRP): A Network Subsystem Architecture for Server Systems**. In Proceedings of OSDI'96: Second Symposium on Operating Systems Design and Implementation, October 1996.

[17] EDWARD, S. S.; SUTARIA, J. **A Study of the Effects of Context Switching and Caching on HTTP Server Performance**. http://www.eecs.harvard.edu/stuart/Tarantula/FirstPaper.html.

[18] BANGA, G.; DRUSCHEL, P. **Measuring the Capacity of a Web Server Under Realistic Loads**. In World Wide Web Journal (Special Issue on World Wide Web Characterization and Performance Evaluation), 1999.

[19] BARFORD, P.; CROVELLA, M. **Critical Path Analysis of TCP Transactions.** In Proceedings of ACM SIGCOMM'2001: Special Interest Group on Computer Communication, Stockholm, Sweden, September 2000.

[20] BARFORD, P.; CROVELLA, M. **Measuring Web Performance in the Wide Area**. In Proceedings of Performance Evaluation Review, August, 1999.

[21] FLOYD, S.; PAXSON, V. **Why We Don't Know How to Simulate the Internet**. In Proceedings of Winter Simulation Conference, Atlanta, GA, December 1997.

[22] FALOUTSOS, M.; FALOUTSOS, P.; FALOUTSOS, C. **On Power-Law Relationships of the Internet Topology**. In Proceedings of ACM SIGCOMM '99: Special Interest Group on Computer Communication, p. 251-262, August 1999.

[23] DRUSCHEL, P.; BANGA, G.; MOGUL, J. C. **A Scalabel and Explicit Event Delivery Mechanism for UNIX**. In Proceedings of USENIX Annual Technical Conference, Monterey, California, June 6-11, 1999.