5 Resultados de simulações

A disciplina proposta, assim como as outras usadas como elementos de comparação, foram simuladas em uma linguagem de programação orientada a eventos discretos [37], chamada SIMSCRIPT [38].

Em uma simulação orientada a eventos discretos, a operação do sistema é representada por uma seqüência cronológica de eventos. Cada evento ocorre em um instante no tempo alterando o estado do sistema. Por exemplo, quando um usuário chega ao sistema, o evento "habilitar tráfego" é acionado promovendo a habilitação dos pacotes deste tráfego e se o escalonador não estiver em serviço, habilita a função do escalonador. Se o pacote for escolhido para ser escalonado o pacote é enviado na rede, senão é inserido na fila do usuário. Essa seqüência de eventos altera constantemente o estado do sistema e ao fim da simulação, estatísticas são geradas para quantificar as grandezas de interesse.

Os componentes da simulação em eventos discretos e a lógica inerente a sequência dos eventos, são os seguintes:

- Relógio: Define a hora corrente da simulação. O relógio anda de forma discreta, pois entre eventos sucessivos, nada acontece que provoque a mudança do sistema;
- Lista de eventos futuros: O sistema mantém uma lista de eventos que ocorrerão em algum momento futuro em relação à hora corrente marcada pelo relógio e em geral estes eventos são agendados por eventos que já ocorreram. A lista de eventos futuros é tipicamente organizada como uma fila de eventos ordenados de forma crescente pelos seus instantes de ocorrência;
- Geradores de números aleatórios: Para a execução da simulação, números aleatórios com distribuições estatísticas específicas precisam

- ser gerados, sendo realizado por meio de "geradores de números pseudo-aleatórios" [38];
- Estatísticas: A simulação tem por objetivo ao seu final de gerar estatísticas de interesse sobre o sistema, cujo objetivo é quantificar as grandezas cujo comportamento médio é necessário conhecer para avaliar o desempenho do sistema. Dentre as grandezas de interesse desçam-se o atraso, o jitter, a taxa de dados e a satisfação dos usuários.

A figura a seguir ilustra os eventos (representados por elipses) de interesse assim como os seus mútuos agendamentos (representados por setas orientadas).

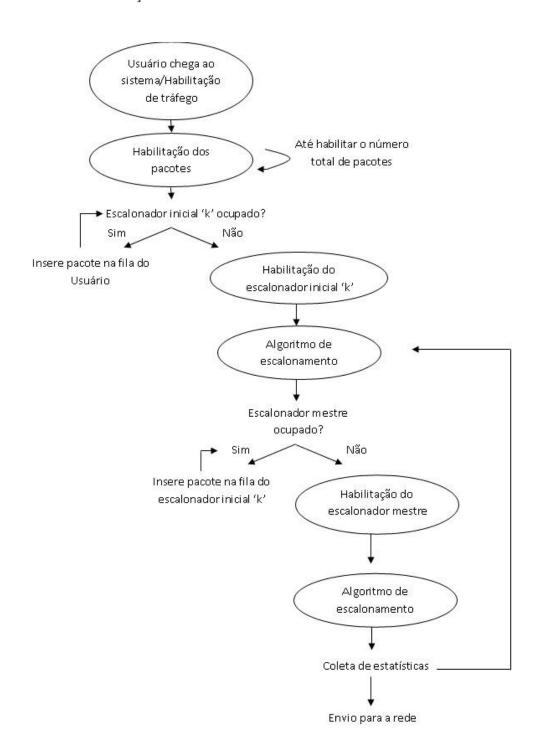


Figura 23 - Seqüência de eventos do sistema

5.1. Cenário de simulação

A definição dos perfis dos usuários criados para o cenário simulado está descrito na tabela abaixo:

	Numero de usuários	Intervalo entre chegadas de usuários (s)*	Delay nominal (s)	Taxa nominal (kbps)	Jitter nominal (s)
FTP 1	10	50	4	2000	1
FTP 2	15	15	4	3000	1
VoIP 1	30	2	0.4	12.4	0.1
HTTP 1	30	2	4	4000	1

^{*} Refere-se à média da função exponencial do intervalo entre chegadas de usuários em segundos

Tabela 6 - Perfil dos Usuários

Já quanto ao perfil geral do sistema, foram adotados os seguintes valores:

Taxa dos Escalonadores iniciais:	5 Mbps
Taxa do escalonador mestre:	10 Mbps
Tempo de simulação:	200 s

Tabela 7 – Perfil do Sistema

Esse cenário é proposto para que se atinja um elevado fator de utilização do sistema com o propósito de simular uma situação de "stress" do sistema onde o desempenho das disciplinas de serviço se torna crítico para o bom funcionamento do sistema.

Esse cenário, independente da disciplina considerada, gerou os seguintes fatores de utilização:

Classe/Tipo de tráfego	Fator de Utilização
FTP 1	86.953 %
FTP 2	83.935 %
VoIP 1	1.819 %
HTTP 1	16.370 %

Tabela 8 - Fatores de utilização

Estes resultados revelam que para escalonadores com taxas de 5 Mbps, foram geradas quantidades de tráfegos suficientes para manter os escalonadores ocupados por tal porcentagem do tempo. As diferenças na utilização vêm do fato de que o pacote de VoIP é 50 vezes menor que o pacote de FTP e aproximadamente 34 vezes menor que o pacote de HTTP, logo o escalonador referente ao VoIP consegue escoar os pacotes com mais rapidez (baixo tempo de processamento) devido ao tamanho reduzido de seus pacotes, obtendo assim, um menor fator de utilização durante o tempo de simulação.

Quanto aos tráfegos de VoIP e HTTP, não é verdade que sempre está ocorrendo transmissão de pacotes, pois há situações em que o usuário de VoIP está em silêncio assim como o usuário de HTTP pode estar apenas lendo a página na WEB.

Já o fator de utilização alcançado pelo escalonador mestre foi de 93.007 %, ou seja, esteve ocupado quase 100% do tempo.

5.2. Análise dos resultados

Após a simulação do sistema, as estatísticas associadas as grandezas de interesse foram gravadas em um arquivo de texto e foram exportadas ao MATLAB para geração dos gráficos necessários a análise dos resultados. As estatísticas antes de serem exibidas, foram submetidas a um filtro de médias

móveis [39] com 1800 amostras para suavização da extrema oscilação dos dados obtidos sem, no entanto perder a visualização das tendências.

5.2.1. Justiça

A definição de uma métrica eficiente de justiça é um trabalho árduo, porém importante. Tal métrica ajudaria no design de uma ampla gama de sistemas, políticas de uso e aplicações tais como servidores web, disciplinas de serviço, aplicações P2P, call centers entre outros. Mesmo assim até os dias de hoje não há uma métrica universalmente aceita.

Os algoritmos "fair-queuing", WFQ[23], SCFQ[24], SFQ[27], são algoritmos que definem justiça através do conceito de "Max-Min Fairness"[36], que em linhas gerais, promovem a alocação de recursos (taxa de dados) aos fluxos de pacotes de forma a maximizar os fluxos com menores taxas de dados.

O seguinte exemplo serve para ilustrar tal conceito:

Considera-se um conjunto de cinco fluxos de pacotes compartilhando um enlace de 10 bit/s onde três fluxos operam em 1 bit/s, um fluxo em 2 bit/s e o fluxo restante em 5 bit/s. De acordo com o conceito acima mencionado, a alocação ocorre no sentido de maximizar os fluxos com menores taxas de dados, logo à todos os fluxos são alocados 1 bit/s maximizando a taxa dos três primeiros fluxos. Agora o fluxo com menor taxa de dados não-maximizada é o fluxo que opera em 2 bit/s, logo é alocado a ele e ao outro fluxo restante mais 1 bit/s, maximizando assim, a taxa desse fluxo (que opera em 2 bit/s). Logo, ainda restam 3 bit/s no enlace e o último fluxo com taxa de dados não maximizada (2 bit/s). Então é alocado a esse ultimo fluxo os 3 bit/s restantes, contabilizando uma taxa maximizada de 5 bit/s, atingindo assim, o "Max-Min Fairness".

Então a justiça desses algoritmos é determinada por fluxo de pacotes tentando manter, para quaisquer par de fluxos, a seguinte igualdade:

$$\left| \frac{w_i(t_1, t_2)}{\rho_i} - \frac{w_j(t_1, t_2)}{\rho_j} \right| \cong 0$$

Onde:

```
ho_i = taxa alocada a sessão i; 
ho_j = taxa alocada a sessão j; 
ho_i = taxa alocada a sessão j; 
ho_i(t_1,t_2) = quantidade de tráfego servido da sessão i em [t_1,t_2]; 
ho_i(t_1,t_2) = quantidade de tráfego servido da sessão j em [t_1,t_2];
```

Entretanto algumas questões podem ser levantadas devido à forma acima de medir justiça entre os usuários. Essa medida pode estar sendo equivocada no seguinte sentido: se um usuário quiser se aproveitar dessa divisão de recursos baseada em fluxo de pacotes, ele pode simplesmente dividir seu fluxo (sessão) em fluxos menores fazendo com que o algoritmo funcione a favor dele reduzindo a oferta de recursos a outros fluxos para prover recursos a esses novos fluxos "menores" do mesmo usuário fazendo assim uma divisão injusta dos recursos. Aplicações P2P geralmente utilizam tais mecanismos de divisão de um fluxo em inúmeros fluxos menores. Essa questão surge, pois os recursos são alocados por fluxo e não por usuários. Por questões de simplificação, no presente trabalho um usuário só pode produzir um único fluxo de pacotes.

Analisando essa questão relacionada à justiça em termos de fluxos de pacotes, fica claro que novas métricas precisam ser desenvolvidas de modo a analisar o comportamento dos usuários sem considerar apenas os fluxos de pacotes.

Com essa idéia em mente é que o presente trabalho segue o caminho da análise da justiça no provimento dos recursos baseado no comportamento dos usuários com o serviço recebido aonde a justiça é relacionada com a comparação entre as satisfações percebidas pelos usuários. Por exemplo, um usuário pode ter uma satisfação muito maior por recurso alocado para diferentes fluxos de pacotes (SMS ou vídeo). Logo diferentes aplicações podem possibilitar diferentes níveis de satisfações com a mesma quantidade de recursos alocados e iguais satisfações podem ser sentidas através de diferentes quantidades de recursos alocados.

Percebe-se então uma nova forma de medir justiça através da satisfação dos usuários com os serviços recebidos. Essas satisfações dos usuários são

medidas através de valores limites aceitáveis quanto a atraso e jitter e quanto à quantidade de recursos alocados.

A figura seguinte mostra a satisfação dos usuários de acordo com cada tipo de serviço.

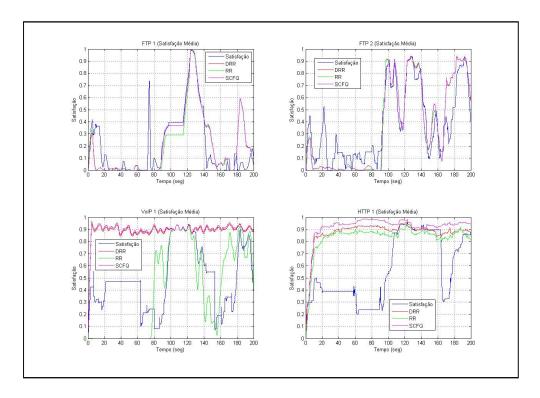


Figura 24 - Satisfação dos usuários

Como se pode ver nos gráficos, houve uma considerável variação na satisfação percebida pelos usuários. O que se pode dizer, é que há momentos em que a disciplina proposta obtém resultados melhores que as outras disciplinas dependendo do tipo de tráfego e, de acordo com essa nova linha de pensamento, obtendo maior justiça nesses casos.

A justiça em relação à disciplina baseada na satisfação obteve melhores resultados em relação às outras disciplinas quando os usuários trafegam a classe de serviço em tempo não-real FTP. Essa classe de serviço foi a classe que mais sobrecarregou o sistema devido à freqüência de requisições de novas sessões e a quantidade de pacotes trafegados por sessão. Já em relação ao tráfego de VoIP, apesar de manter os usuários com satisfação maior que 50% em boa parte do tempo de simulação do sistema, a disciplina proposta teve um desempenho razoável em relação à disciplina "Round Robin" porém insatisfatória em relação

às demais. Em relação ao tráfego de HTTP, a disciplina proposta mostrou um menor grau de justiça em relação às outras disciplinas consideradas.

A satisfação é calculada baseada nas métricas de qualidade de serviço de cada tipo de tráfego, logo seria adequado que o gráfico de cada satisfação ilustrasse também as métricas de qualidade de serviço de acordo com o tipo de tráfego, facilitando sobremaneira as análises dos resultados. Este é o objetivo da subseção seguinte.

5.2.2. Métricas de Qualidade de serviço

As métricas de qualidade de serviço utilizadas para cálculo da satisfação são atraso, jitter e taxa de transmissão de dados. Outra métrica que surge como importante na simulação do sistema é o tamanho médio da fila dos pacotes dos usuários pertencentes a um tipo de tráfego e o tamanho médio da fila pertencente ao escalonador de cada tipo de tráfego.

Esses resultados são mostrados em conjunto com a satisfação para melhor entender o que acontece no decorrer da simulação do sistema.

5.2.2.1. FTP 1

Os gráficos referentes à satisfação, atraso, jitter e taxa de transmissão dos dados são mostrados a seguir. Depois é feita a análise da satisfação baseada nas métricas de QoS.

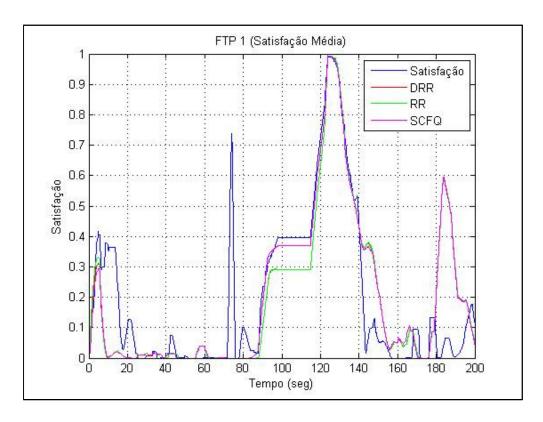


Figura 25 - Satisfação dos usuários de FTP 1

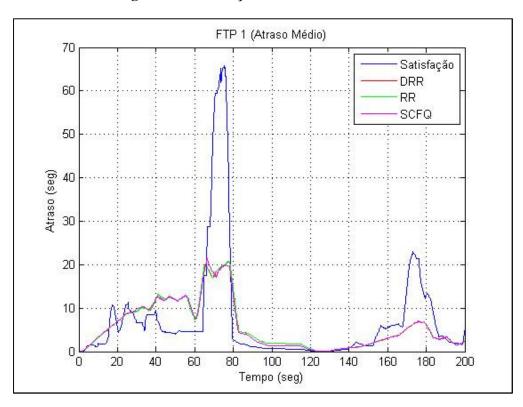


Figura 26 – Atraso dos usuários de FTP 1

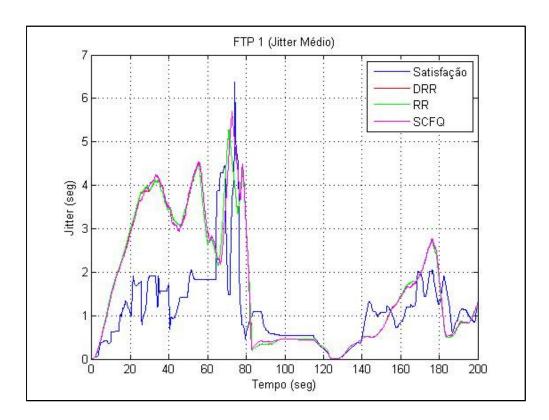


Figura 27 - Jitter dos usuários de FTP 1

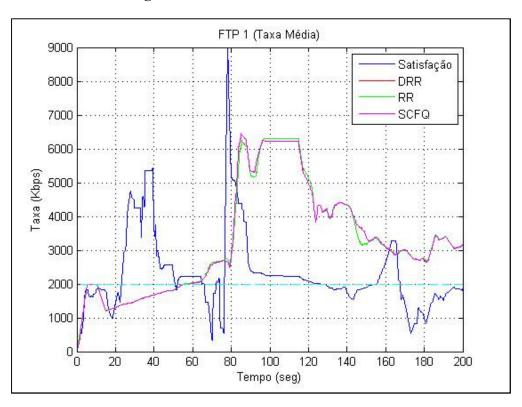


Figura 28 – Taxa de transmissão dos usuários de FTP 1

A satisfação da disciplina proposta (azul) é superior às outras desde o começo da simulação até o instante aproximado de 80 segs. Nesse intervalo de tempo, tanto o atraso quanto o jitter se mantêm em níveis menores do que os correspondentes das outras disciplinas quando então o atraso aumenta consideravelmente, chegando ao ponto em que esse tráfego necessite de priorização por parte dos escalonadores para que seus pacotes sejam escoados, as métricas de qualidade de serviço re-estabelecidas de acordo com os valores limites aceitáveis e as satisfações dos usuários melhoradas. De acordo com essa priorização, a taxa de transmissão chega ao pico de 9000 kbps em torno do instante "80" na tentativa de os objetivos anteriormente mencionados.

A figura seguinte descreve a evolução temporal do tamanho da fila de pacotes dos usuários e mostra que nesse instante ("80s") o escalonador mestre fica sobrecarregado, provocando assim atrasos nos pacotes de FTP 1.

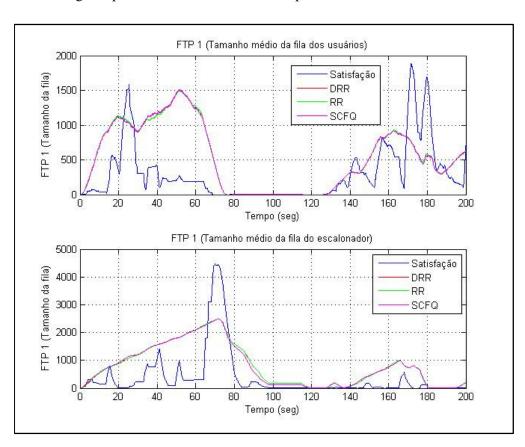


Figura 29 - Tamanho médio das filas de pacotes dos usuários de FTP 1

A partir do instante "160" os usuários voltam a diminuir sua satisfação, pois ocorre um aumento no atraso e jitter dos pacotes devido à sobrecarga do escalonador inicial "1" pertencente ao FTP 1.

A disciplina "Round Robin" e a "Deficit Round Robin" obtêm desempenho semelhantes, pois enquanto a primeira serve um pacote por vez, a segunda serve dois pacotes fazendo com que a diferença de desempenho das duas se torne imperceptível.

A disciplina "Self-Clocked Fair Queuing" obteve um desempenho razoável a partir do instante "90" (aproximadamente), pois manteve os usuários entre os mais satisfeitos quando comparado às outras disciplinas, manteve as maiores taxas de transmissão, os menores atrasos médios e usuários alcançando picos de 100% de satisfação em torno do instante "120".

É interessante notar que nos intervalos de tempo onde os usuários estiveram mais satisfeitos foram justamente os intervalos em que o tamanho das filas tanto dos usuários quanto do escalonador inicial "1" esteve próximo de zero.

5.2.2.2. FTP 2

Os tráfegos FTP 1 e FTP 2 diferem no que diz respeito as médias do intervalo entre chegadas de usuários e dos tamanhos dos arquivos trafegados. No FTP 2, é considerada uma média do intervalo entre chegadas de usuários de 15 segundos e uma média de 500 KB por arquivo trafegado. A presente análise segue da interpretação dos gráficos referentes à satisfação, atraso, jitter e taxa de transmissão de dados apresentados a seguir.

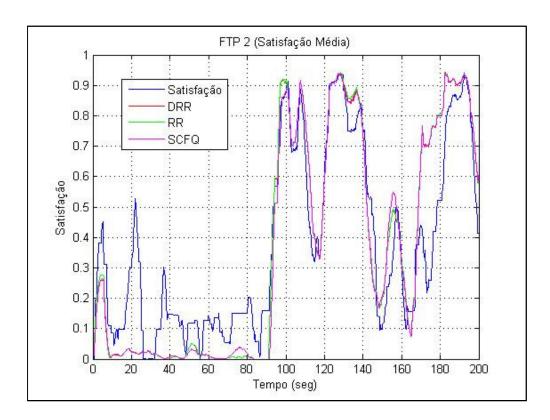


Figura 30 – Satisfação dos usuários de FTP 2

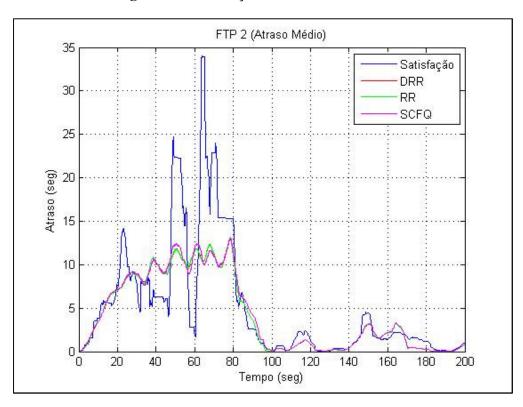


Figura 31 – Atraso dos usuários de FTP 2

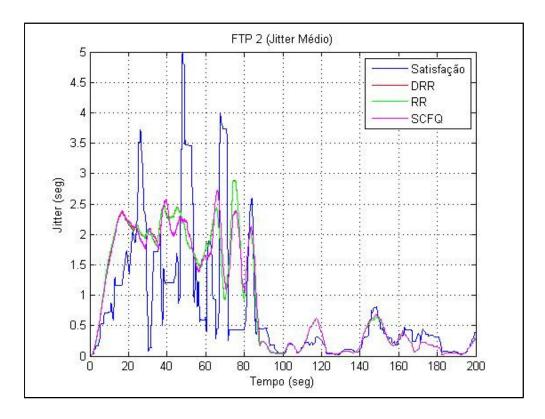


Figura 32 – Jitter dos usuários de FTP 2

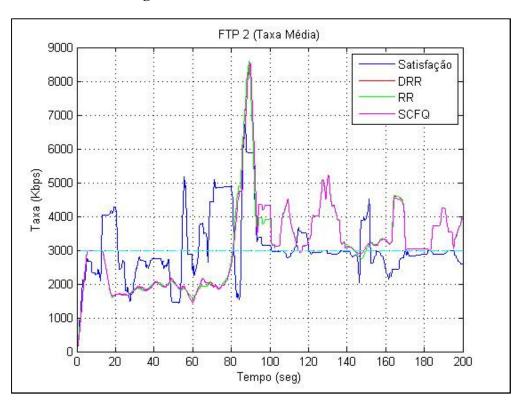


Figura 33 – Taxa de transmissão dos usuários de FTP 2

A satisfação dos usuários de FTP 2 com a disciplina proposta foi a maior dentre todas as disciplinas simuladas, desde o início de simulação do sistema até o instante de 90 segundos. Tanto o atraso quanto o jitter tiveram variações maiores na disciplina proposta quando comparadas com as demais.

O que se pode deduzir ao analisar o comportamento do tráfego FTP 2, em linhas gerais, do começo da simulação até o instante "90", é que como em um determinado instante o atraso e jitter estavam em níveis maiores levando a satisfação a níveis baixos, o tráfego em questão era priorizado acarretando o aumento da taxa de transmissão pois os pacotes estavam sendo escoados, fazendo com que suas satisfações aumentassem e atraso e jitter diminuíssem, quando então o trafego deixava de ser priorizado. Em um momento seguinte, como este tráfego não estava sendo priorizado com a mesma intensidade, a taxa caía, atraso e jitter aumentavam e a satisfação diminuía. Então depois de chegar num nível que a priorização era crucial, o tráfego era novamente priorizado melhorando as métricas e satisfações dos usuários.

De acordo com a figura 34, foi nesse intervalo de tempo (0-90s) em que tanto o escalonador inicial "2" quanto o mestre ficaram sobrecarregados de pacotes FTP 2. Após esse intervalo, as filas não permaneceram com muitos pacotes enfileirados.

A partir do instante "90", as disciplinas tiveram desempenho semelhante onde a disciplina proposta teve um desempenho um pouco abaixo das demais.

A diferença no desempenho das disciplinas "Round Robin" e "Déficit Round Robin" é quase imperceptível, pois enquanto que a primeira envia um pacote por vez, a segunda envia dois pacotes.

A disciplina "Self-Clocked Fair Queuing" obtém um desempenho semelhante ao "Round Robin" e "Déficit Round Robin" pelo fato de priorizar com mais intensidade os outros tipos de tráfego em relação a este.

A próxima figura mostra a evolução temporal do tamanho das filas dos usuários e a do escalonador inicial "2". Os tamanhos das filas de pacotes dos usuários tiveram uma média alta mostrando os intervalos de tempo em que o tráfego de FTP 2 esteve mais intenso em relação aos demais, sobrecarregando

assim, o funcionamento do escalonador inicial "2" e do escalonador mestre. O gráfico referente à figura 34, diz respeito ao tamanho médio da fila de pacotes do escalonador inicial "2" e mostra que quando as satisfação dos usuários FTP 2 foi a maior entre todas as disciplinas (0 - 90 s), o tamanho médio das filas da disciplina proposta foi sempre menor que os das outras disciplinas consideradas. A partir desse ponto (90 s), as disciplinas priorizam os usuários de forma parecida com pouca variação no tamanho médio das filas do escalonador inicial "2".

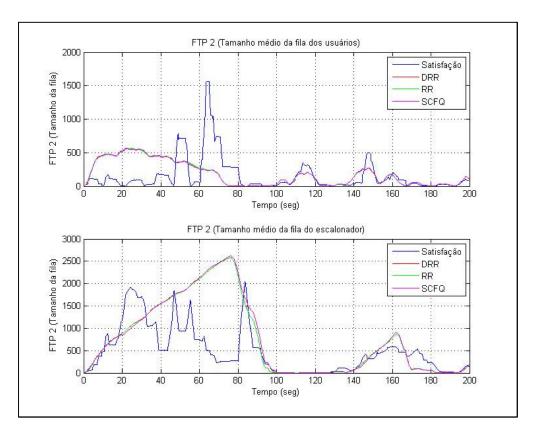


Figura 34 - Tamanho médio das filas de pacotes dos usuários de FTP 2

5.2.2.3. VoIP 1

A satisfação dos usuários de VoIP foi a que mais apresentou desempenho significativamente diferentes em função das disciplinas. Os usuários de VoIP são usuários que trafegam pacotes em tempo real e esse tipo de tráfego apresenta diferentes requisitos de desempenho em relação à FTP e HTTP. Esse tipo de tráfego é mais sensível em relação a atrasos e jitter, e por isso a disciplina

simulada precisa oferecer garantias de atraso e jitter mais estritos para não tornar a comunicação ininteligível.

A forma como a disciplina proposta trata essa questão é através da definição de valores limites aceitáveis mais restritos tanto de jitter quanto de atraso. Esses valores são mostrados na tabela "7".

Seguem então, os gráficos da satisfação, atraso, jitter e taxa de transmissão de dados dos usuários de VoIP para análise.

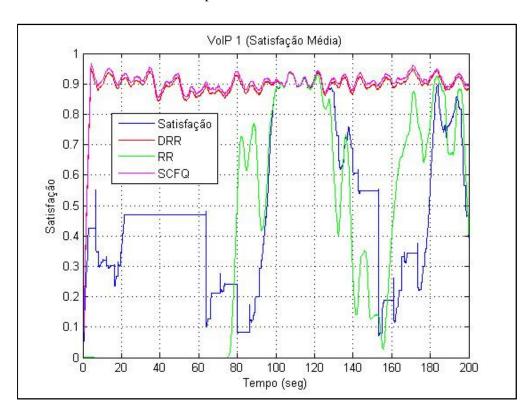


Figura 35 - Satisfação dos usuários de VoIP 1

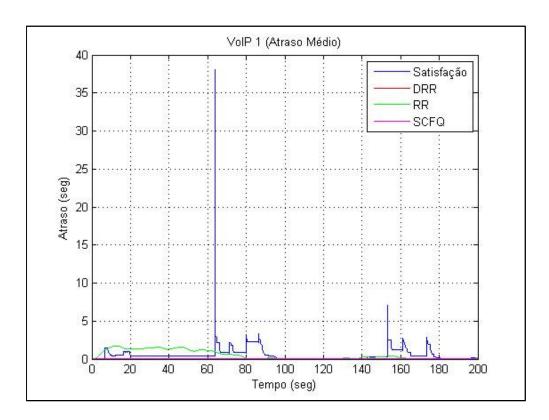


Figura 36 – Atraso dos usuários de VoIP 1

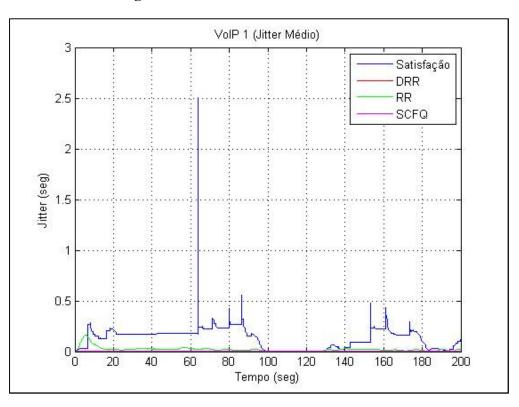


Figura 37 – Jitter dos usuários de VoIP 1

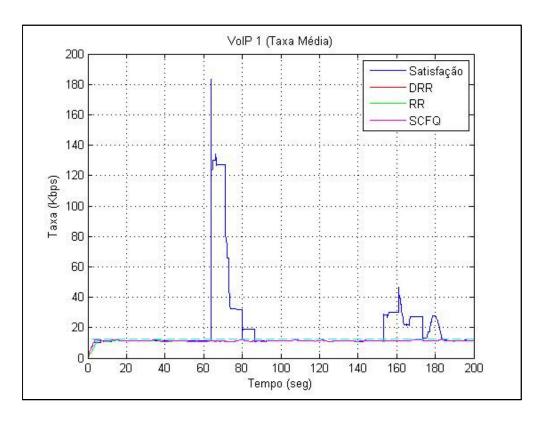


Figura 38 – Taxa de transmissão de dados dos usuários de VoIP 1

A satisfação dos usuários da disciplina proposta apresentou valores melhores que o "Round Robin" até o instante "70". A partir desse instante ocorre um atraso da ordem de 30 segundos fazendo com que a satisfação diminua. Ocorre então a priorização, elevando a sua taxa de transmissão até que o atraso e o jitter se restabeleçam. Posteriormente um leve atraso ocorre, fazendo com que a satisfação dos usuários sofra uma depressão no instante "150", quando então ela é novamente priorizada até que os níveis das métricas de QoS se tornem aceitáveis.

A disciplina "Déficit Round Robin" obteve um ótimo desempenho, pois ela foi concebida para ofertar janelas de transmissão iguais para todos os tráfegos e como o tamanho do pacote VoIP é algo em torno de 30 vezes menor que os outros tráfegos, muitos pacotes são encaixados nessa janelas fazendo com que o tráfego de VoIP obtenha mais serviço por vez que o escalonador visita sua fila. No presente trabalho a janela de transmissão esta configurada para "3000 bytes", e como os pacotes de VoIP tem tamanho de 30 bytes, enquanto o FTP "encaixa" dois pacotes nessa janela e o HTTP três pacotes, o VoIP "encaixa" 100 pacotes, promovendo o ótimo desempenho.

A disciplina "SCFQ" também obteve um ótimo resultado, pois conseguiu manter todas as métricas de qualidade de serviço em níveis menores aos máximos aceitáveis.

A disciplina "Round Robin" variou a satisfação de acordo com suas métricas de qualidade de serviço. Essa disciplina manteve os usuários com satisfação nula até o instante 70 (aprox.), pois manteve um nível de atraso constante superior ao máximo aceitável. Depois, o tráfego é priorizado aumentando sua satisfação até em torno de "130" segundos e voltando a cair devido ao aumento no atraso dos pacotes, dessa vez, entretanto, não ao ponto de manter os usuários totalmente insatisfeitos. A figura 39 mostra os níveis de atraso e jitter no intervalo entre 0 e 100s, ilustrando a razão dos usuários estarem com satisfação nula.

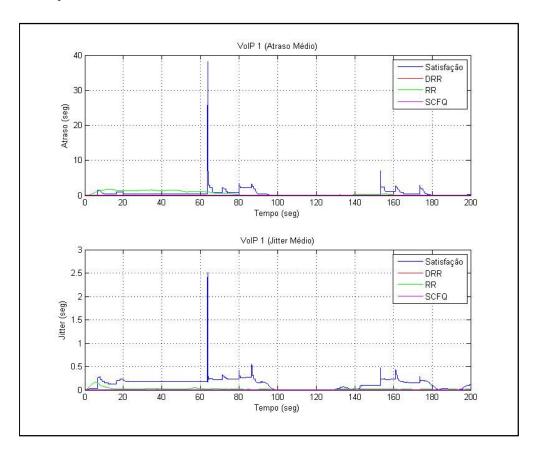


Figura 39 - Delay e Jitter entre 0 e 100 s dos usuários de VoIP 1

O valor limite aceitável de atraso para que não haja comprometimento na comunicação é de 0.4 s [14]. Logo se percebe que durante esse intervalo de tempo, o "Round Robin" manteve um atraso em níveis três vezes maiores que o

aceitável. O jitter apresentou valores aceitáveis, pois o atraso não variou muito e sim apenas permaneceu em níveis maiores que os aceitáveis.

A figura 40 mostra que os usuários tiveram suas filas com mais pacotes esperando por serviço quando o "Round Robin" é simulado e evidencia o momento em que a disciplina proposta apresenta uma queda de desempenho em torno dos 70 segundos.

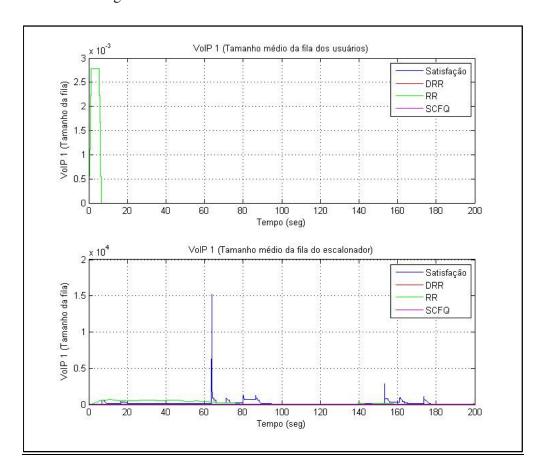


Figura 40 - Tamanho médio das filas de pacotes dos usuários de VoIP 1

5.2.2.4. HTTP 1

O tráfego de HTTP não é tão sensível a atraso e jitter como o VoIP, porém o fato de um usuário tentar surfar na Web e sofrer o mínimo atraso que for, já é o suficiente para ficar insatisfeito e possivelmente trocar de provedora de serviços de telecomunicações.

Os gráficos referentes à satisfação, atraso, jitter e taxa de transmissão de dados são mostrados a seguir.

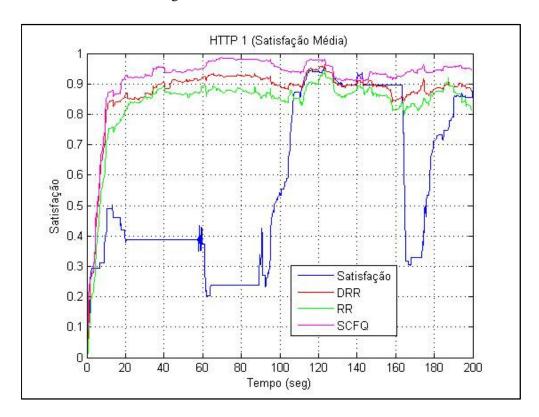


Figura 41 – Satisfação dos usuários de HTTP 1

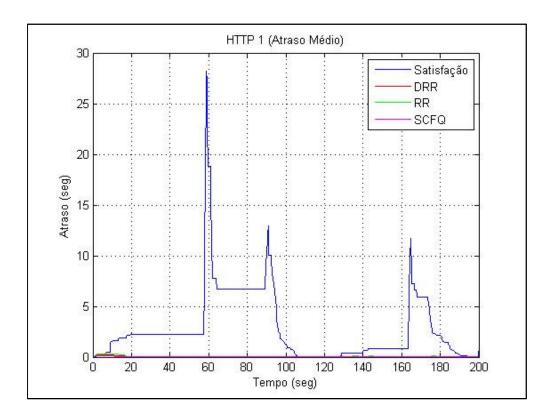


Figura 42 – Atraso dos usuários de HTTP 1

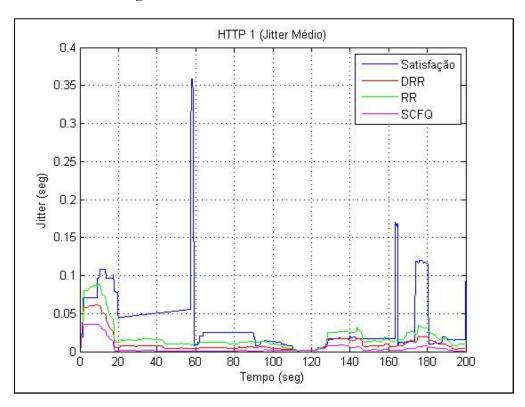


Figura 43 – Jitter dos usuários de HTTP 1

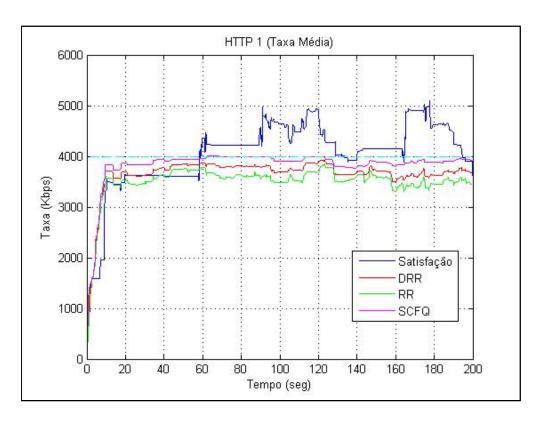


Figura 44 - Taxa de transmissão de dados dos usuários de HTTP 1

A satisfação média da disciplina proposta teve o seu pior desempenho, com uma melhora significativa a partir do instante "100" depois caindo novamente. Esse fato é evidenciado pelas métricas de qualidade de serviço. A taxa de transmissão ficou um pouco acima da média por causa dos momentos em que a priorização era crucial, fazendo com que a taxa alcançasse picos de transmissão em torno de 5 Mbps.

As demais disciplinas tiveram desempenhos bem parecidos, sendo que a "SCFQ" conseguiu manter uma taxa de transmissão maior que as demais. Na disciplina proposta os pacotes sofrem um atraso considerável em torno de 60 segundos de simulação por não estarem recebendo serviço. Assim atinge-se um limite onde os usuários permanecem tão insatisfeitos até que o escalonador decide priorizar aumentando a taxa de transmissão. Com estas sucessíveis priorizações, as métricas chegam a níveis aceitáveis, repetindo-se este processo em torno do instante "160".

De acordo com a figura 45 (referente ao tamanho médio das filas de pacotes), percebe-se que o escalonador inicial "4" pertencente ao HTTP 1

utilizando a disciplina proposta consegue ofertar serviço com mais qualidade do que as demais disciplinas. Já o escalonador mestre já não prioriza tanto o trafego de HTTP 1, pois como a satisfação é na média maior que a dos outros tipos de tráfego, o escalonador mestre entende que esses usuários podem esperar um pouco mais para receber serviço dando prioridade aos outros tipos de tráfego e aumentando o tamanho médio da fila dos pacotes.

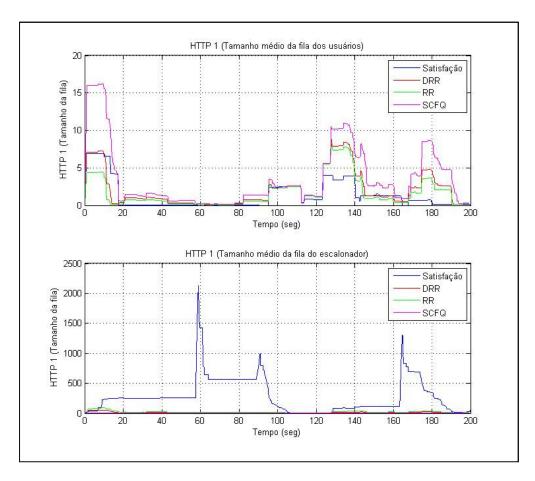


Figura 45 - Tamanho médio das filas de pacotes dos usuários de HTTP 1

5.2.3. Complexidade de implementação

A motivação para a determinação da complexidade de implementação dos algoritmos estudados advêm do seguinte questionamento: "Se o numero de usuários /sessões aumentar, qual será o fator de acréscimo no tempo de execução

dos algoritmos estudados? E quais são as implicações disso?". Este é um problema denominado de "escalabilidade" do sistema.

A complexidade no tempo de execução de um problema pode ser expressa pelo número de operações necessárias para resolver uma instância do problema em função da quantidade dos dados de entrada. Se, por exemplo, uma instância do problema tem "n" dados de entrada e pode ser resolvido em " n^2 " operações, pode-se dizer que o problema tem complexidade no tempo de " n^2 ". A notação aqui usada é chamada de "Big O notation" [41], onde "O" significa ordem de crescimento de um algoritmo. Logo no exemplo acima o problema apresenta complexidade de $O(n^2)$.

No contexto do presente trabalho, esta notação serve para determinar um valor do número de operações necessárias que o algoritmo deve realizar para decidir o usuário/sessão e a classe/tipo de tráfego que será escalonado baseado no número de usuários/sessões e classe/tipos de tráfegos ativos (filas não-vazias). Os "k" escalonadores iniciais possuem um conjunto de entrada de " n_k " filas de usuários /sessões. O escalonador mestre possui um conjunto de "k" filas de escalonadores iniciais. Lembrando que uma classe/tipo de tráfego se refere a um escalonador inicial e o cálculo da complexidade é feito para o pior caso, ou seja, todos os usuários/sessões e escalonadores iniciais ativos (filas não-vazias).

analisando modo algoritmos realizam Então, o como os escalonamentos, percebe-se que os algoritmos das disciplinas "Frame-Based" não realizam tantas operações para escalonar quanto numero de usuários/sessões nem de escalonadores iniciais. Logo sua complexidade de implementação está na ordem de O(1) para os dois escalonamentos realizados. Isso quer dizer que independente do número de usuários/sessões e escalonadores iniciais ativos, o algoritmo realiza uma operação nos escalonadores iniciais e uma no escalonador mestre. No caso dos algoritmos "Frame-Based", o que o algoritmo faz é checar se há pacotes na fila ou se a janela de transmissão comporta mais pacotes. Então decide o que fazer: escalona ou passa o processamento para a próxima fila.

Já os algoritmos realizados pelas disciplinas "Sorted Priority", como o próprio nome diz, designa prioridades para os usuários/sessões que estão competindo para serem escalonados. No caso da "Disciplina baseada na

Satisfação", a insatisfação é atualizada apenas para o usuário que trafegou o pacote na rede e essa insatisfação é utilizada para o escalonamento do "k" escalonador inicial, logo independente do número de usuários/sessões ativos. Já o escalonador mestre realiza tantas operações quanto numero de filas dos "k" escalonadores iniciais, pois a insatisfação média por escalonador é sempre recalculada para o escalonamento. Logo sua complexidade fica em torno de O(1) para os escalonadores iniciais e O(k) para o escalonador mestre.

Já a disciplina "Self-Clocked Fair Queuing", tem complexidade de $O(\log_2(n_k))$ para os escalonadores inicias e $O(\log_2(k))$ para o escalonador mestre. Esse resultado é provado em [42], e advêm do fato de que essa disciplina calcula o instante virtual de fim de serviço do pacote atual, tanto dos usuários/sessões quanto dos escalonadores iniciais.

A tabela a seguir apresenta um resumo das disciplinas de serviço e suas complexidades numéricas.

Disciplina	Complexidade		
	Escalonadores Iniciais	Escalonador Mestre	
Satisfação	O(1)	O(k)	
Self-Clocked Fair Queuing	$O(\log_2(n_k))$	$O(\log_2(k))$	
Round Robin	O(1)	O(1)	
Deficit Round Robin	O(1)	O(1)	

Tabela 9 - Complexidade das disciplinas de serviço

A complexidade de implementação se torna relevante quando as taxas de transmissão chegam à casa do GB/s e os pacotes precisam ser escalonados na ordem de nanossegundos. Imaginando o caso em que há cerca de mil usuários competindo por serviço e que a disciplina precise calcular as prioridades para cada usuário para decidir quem receberá serviço e esses usuários estejam transmitindo na ordem de vários MB/s, fica evidente que o tempo gasto para fazer todos esses cálculos impacta diretamente no desempenho da disciplina.

Devido à sua baixa complexidade de implementação, a disciplina "Deficit Round Robin" e sua variação "Modified Deficit Round Robin" são amplamente usadas em roteadores e switches da empresa CISCO.