

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ESPECIALIZAÇÃO EM TELEINFORMÁTICA E REDES DE
COMPUTADORES**

DANÚBIA RAVENA SILVA BARBOSA

**ESTUDO COMPARATIVO DOS ESCALONADORES DE PACOTES
QUE GERAM QoS EM UMA REDE DE DADOS**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA
2011

DANÚBIA RAVENA SILVA BARBOSA

**ESTUDO COMPARATIVO DOS ESCALONADORES DE PACOTES
QUE GERAM QoS EM UMA REDE DE DADOS**

Trabalho de Conclusão de Curso de pós-graduação, apresentado à disciplina Trabalho de Diplomação, do curso de especialização em Teleinformática e Redes de Computadores da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de especialista.
Orientador: Prof. Dr. Kleber Nabas

CURITIBA
2011



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CURITIBA
CURSO DE ESPECIALIZAÇÃO EM TELEINFORMÁTICA E
REDES DE COMPUTADORES
DISCIPLINA DE TRABALHO DE DIPLOMAÇÃO

TERMO DE APROVAÇÃO

DANÚBIA RAVENA SILVA BARBOSA

ESTUDO COMPARATIVO DOS ESCALONADORES DE PACOTES QUE GERAM QoS EM UMA REDE DE DADOS

Este Trabalho de Diplomação foi considerado adequado como cumprimento das exigências legais do currículo da Especialização em Teleinformática e Redes de Computadores, da Universidade Tecnológica Federal do Paraná – Campus de Curitiba, para banca formada dos seguintes professores:

A handwritten signature in blue ink, appearing to be 'Kleber Nabas', is written over a horizontal line.

Orientador: Prof. Dr. Kleber Nabas

A handwritten signature in blue ink, appearing to be 'Walter Godoy Júnior', is written over a horizontal line.

Coordenador: Prof. Dr. Walter Godoy Júnior

Nota: 8,0 (OITO INTEIROS)

AGRADECIMENTOS

Primeiramente agradeço a Deus por mais uma oportunidade em minha vida de estar cumprindo mais uma etapa de estudos. Sei que não é fácil, mas se tudo na vida fosse fácil jamais teria graça.

Agradeço a minha mãe por sempre estar ao meu lado me ajudando em cada dificuldade em cada passo que eu dou na minha vida.

Agradeço a minha prima Bruna que veio morar comigo em Curitiba, para enfrentar mais uma batalha em nossas vidas; Agradeço também a minha Tia Ângela que também veio morar aqui em Curitiba conosco, obrigada Tia por tudo, obrigada por sempre cuidar de mim!

Se eu for agradecer a todos que estão presentes na minha vida me ajudando e me apoiando aqui não haveria esse espaço. Mas eu me lembro de todos e com certeza esses tem um lugar especial no meu coração.

Agradeço a todos os professores da Pós, foram momentos pequenos, mas que valeram à pena.

Agradeço a Deus pelos colegas e amigos que fiz na Pós, afinal a vida sem amigos não é nada.

Obrigada Senhor por mais uma etapa concluída, e vamos que vamos para mais uma jornada da vida.

RESUMO

Este trabalho tem por objetivo apresentar um estudo comparativo entre escalonadores de pacotes que geram qualidade de serviço (QoS) em uma rede de dados, o mesmo apresenta especificações da qualidade geradas dentro da rede dados onde é gerenciado o fluxo de pacotes e tráfego que utiliza um conjunto de mecanismos de escalonamento. A QoS é a arquitetura que busca garantir a qualidade no tráfego e serviços em redes de computadores. O controle de tráfego é o conjunto de mecanismos de escalonamentos que vêm através de pacotes dentro de um *router*. Roteadores são elementos centrais obter a qualidade necessária nos pacotes que trafegam na rede até chegar ao seu destino final.

Palavras-chave: Escalonamento. Pacotes. Qualidade. Tráfego.

ABSTRACT

This work intends to present a comparative study of packages schedulers that generate quality of service (QoS) in a data network, the same quality specifications has generated within the network where data is managed and the packet stream from traffic which uses a set of scheduling mechanisms. The QoS architecture is seeking to ensure quality services and traffic in computer networks. Traffic control is the set of mechanisms that come through escalation of packets within a router. Routers are central to obtaining the necessary quality in the packets that travel over the network through to reach their final destination.

Keywords: Scheduling. Packages. Quality. Traffic.

SUMÁRIO

RESUMO.....	4
ABSTRACT	5
SUMÁRIO	6
LISTA DE FIGURAS	8
LISTA DE SIGLAS E ABREVIATURAS	11
INTRODUÇÃO	12
1. QUALIDADE DE SERVIÇO (QoS)	15
1.1 COMO OBTER QoS	17
2. ESCALONAMENTO DE PROCESSOS	21
2.1 Níveis de Escalonamento	26
2.2 Escalonamentos Preemptivo e Não Preemptivo	28
3. ALGORITMOS DE ESCALONAMENTO	30
3.1 ESCALONAMENTO FIFOQ (<i>First In First Out Queueing</i>)	31
3.2 ESCALONAMENTO FQ (<i>Fair Queueing</i>)	33
3.3 ESCALONAMENTO CQ (<i>Custom Queueing</i>)	34
3.4 ESCALONAMENTO GPS (<i>Generalized Processor Sharing</i>)	34
3.5 ESCALONAMENTO CBQ (<i>Class Based Queueing</i>).....	35
3.6 ESCALONAMENTO WFQ (<i>Weighted Fair Queueing</i>).....	36
3.7 ESCALONAMENTO WRR (<i>Weighted Round Robin</i>)	37

3.8 ESCALONAMENTO PQ (<i>Priority Queueing</i>).....	38
3.9 ESCALONAMENTO HPF (Highest Priority First)	38
4. O ESCALONADOR WFQ	40
4.1 ASPECTOS ANALÍTICOS DO WFQ	45
4.2 EXEMPLO DO FUNCIONAMENTO WFQ	48
5. CONCLUSÃO	53
6. REFERENCIAL BIBLIOGRÁFICO	54

LISTA DE FIGURAS

Figura 1: Tabela de Vazão Típica de Aplicação em Rede	17
Figura 2: Aplicação de QoS.....	20
Figura 3: Estados sucessivos de um processo no sistema	22
Figura 4: Conceitos básicos de um Escalonamento do Processador.....	25
Figura 5: Explicação para o programa da figura anterior	25
Figura 6: Escalonamento e transições de estado e processos	27
Figura 7: Escalonador FIFOQ (BALLIACHE,2003)	32
Figura 8: Escalonador FQ (BALLIACHE,2003)	33
Figura 9: Custom Queueing	34
Figura 10: Escalonamento CBQ.....	35
Figura 11: Escalonador WFQ (BALLIACHE,2003)	36
Figura 12: Exemplo de compartilhamento de tráfego de voz e dados utilizando WFQ	37
Figura 13: Escalonamento HPF	38
Figura 14: Escalonamentos de Pacotes WFQ (MELO, 2001)	42
Figura 15: Exemplo de funcionamento do WFQ.....	42

Figura 16: Quadro 1 - Pseudocódigo ao algoritmo de escalonamento WFQ	43
Figura 17: Configuração do Escalonador WFQ, compartilhando duas fontes com 90% e 10%.....	44
Figura 18: Largura de Banda utilizada pelos fluxos 1 e 2 da Figura 16.....	45
Figura 19: Modelo de um nó com o escalonador WFQ (DEKERIS, ADOMKUS, BUDNIKAS,2006).....	46
Figura 20: Ciclo 1 - WFQ.....	48
Figura 21: Ciclo 2 - WFQ.....	49
Figura 22: Ciclo 3 - WFQ.....	49
Figura 23: Ciclo 4 - WFQ.....	50
Figura 24: Ciclo 5 - WFQ.....	50
Figura 25: Ciclo 6 - WFQ.....	51
Figura 26: Ciclo 7 - WFQ.....	52
Figura 27: Ciclo 8 - WFQ.....	52

LISTA DE SIGLAS E ABREVIATURAS

QoS	<i>Quality of Service</i>
UCP	Unidade Central de Processamento
PWB	Parte da área de um processo
RISC	<i>Reduced Instruction Set Computer</i>
MPLS	<i>Multiprotocol Label Switching</i>
IETF	<i>Internet Engineering Task Force</i>
RFC	<i>Request For Comments</i>
FIFOQ	<i>First In First Out Queueing</i>
FQ	<i>Fair Queueing</i>
CQ	<i>Custom Queueing</i>
GPS	<i>Generalized Processor Sharing</i>
CBQ	<i>Class Based Queueing</i>
WFQ	<i>Weighted Based Queueing</i>
WRR	<i>Weighted Round Robin</i>
PQ	<i>Priority Queueing</i>
INTSERV	Serviços Integrados
DIFFSERV	Serviços Diferenciados
CPU	Unidade Central de Processamento
PWB	Parte da área de um processo
RISC	<i>Reduced instruction set computing</i>
MP	Memória Principal
BRR	<i>Bit-by-bit-Round-Robin</i>
IP	<i>Internet Protocol</i>
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
PGPS	<i>Packet Generalized Processor Sharing</i>

INTRODUÇÃO

Hoje há crescente demanda por serviços de dados e multimídia de alta velocidade que se alastra entre vários biótipos de clientes. E este mercado que cresce cada dia mais vem em busca de tecnologias inovadoras que são capazes de oferecer sempre mais e mais qualidade aos usuários que utilizam essa grande nuvem, chamada rede de dados.

Há uma grande preocupação para com a qualidade dos serviços oferecidos em meio tanta tecnologia; e é nesse ponto que entraremos com o nosso estudo sobre a qualidade dos escalonadores de pacotes.

O termo QoS¹ significa ter a capacidade de fornecer um serviço conforme as exigências propostas pela rede, ou seja, em tempos de resposta quantitativas (para que assim não aconteça a perda de pacotes) e a banda concorrida. A QoS deve garantir um nível aceitável de perda de pacotes nas redes, mas, mesmo com tantos termos para uma qualidade de rede, nada garante uma rede sem congestionamentos.

E então temos os chamados “mecanismos de QoS”, estes que mostram diferentes fluxos de rede e reservam uma parte da banda concorrida para os que necessitam de um serviço contínuo, sem perdas ou cortes.

¹ Quality of Service

Com base no estudo de Kleber Nabas², podemos ter uma definição sobre escalonadores de pacotes (2009, p. 4),

O escalonador de pacotes gerencia as filas de diferentes fluxos de pacotes utilizando um conjunto de mecanismos de escalonamento. O escalonador é implementado no local onde os pacotes estão enfileirados e pode implementar funções de policiamento de tráfego, que assegura que um usuário não viole as características prometidas. A função básica do escalonador é reordenar a saída das filas e isso pode ser realizado de diversas maneiras. A forma mais simples, porém, é a estratégia de prioridades, onde os pacotes são ordenados pela sua prioridade e os que possuem prioridades mais altas deixam a fila antes. Outra forma é o escalonamento denominado cíclico (round-robin), que atribui acessos distintos aos pacotes com classes diferentes para compartilhar um enlace.

Alguns escalonadores mais conhecidos são:

- _ Escalonador priorizado PQ (Priority Queueing);
- _ Escalonador justo ponderado WFQ (Weighted Fair Queueing);
- _ Escalonador baseado em classes CBQ (Class Based Queueing).

O Escalonador de pacotes CBQ, também conhecido como escalonador hierárquico, está baseado na criação de classes ou hierarquias para diversos tipos de fluxos que um roteador deverá lidar. Para cada uma destas classes é alocada uma parte da banda disponível e também atribuída uma prioridade. [1].

O potencial de um escalonador de pacotes é essencial para redes de dados; Estes suportam a integração de serviço tornando-se assim necessário a providência de QoS diferenciada por categoria de serviço para poder oferecer garantias de desempenho a aplicações.

O capítulo 1 vem fazer uma breve explicação sobre o que é qualidade de serviço em uma rede de dados, como funciona, suas aplicações e parâmetros que buscam a garantia do tráfego e serviços dentro da rede.

O capítulo 2 descreve sobre Escalonamento de Processos, seu conceito básico e objetivo logo no princípio dos Sistemas Operacionais onde estão alocados, mostrando que é um conceito principal para o uso da política adequada de escalonamento.

O capítulo 3 explica os algoritmos de escalonamentos, estes que são componentes essenciais para atingir a qualidade de serviço em uma rede de dados.

² Prof. Dr. Kleber Nabas – Universidade Tecnológica Federal do Paraná (UTFPR)

De seguida vem mostrando e exemplificando os mecanismos de escalonamentos, tais quais são: FIFOQ, FQ, CQ, GPS, CBQ, WFQ, WRR, PQ e HPF.

Já para capítulo 4 é demonstrado mais a fundo o escalonador WFQ, que será o escolhido nesse trabalho como escalonador de pacotes modelo, pois neste pode-se fazer uma diferenciação dos fluxos e assim prover a qualidade de serviço em diferentes aplicações.

Na conclusão que será o último capítulo desenvolvido neste trabalho, vai ser fechado com a análise geral, salientando a importância de se escolher um escalonador de pacotes, que controle a qualidade dos fluxos na rede e então possa assim prover a QoS da mesma.

1. QUALIDADE DE SERVIÇO (QoS)

Quality of Service (QoS), é o serviço que vem garantir a qualidade do tráfego e serviços nas redes de computadores.

“Por definição, a Qualidade de Serviço (Quality of Service – QoS) de uma rede é garantida pelos componentes da rede e equipamentos utilizados, estando baseada em um mecanismo fim-a-fim de garantir a entrega das informações e que deve atuar na comunicação dos equipamentos envolvidos visando o controle dos parâmetros de Qualidade de Serviço. Mas, afinal, o que é a Qualidade de Serviço?

Num primeiro momento, o termo "Qualidade de Serviço" pode ser entendido como sendo um requisito das aplicações para a qual exige-se que determinados parâmetros (atrasos, vazão, perdas, etc) estejam dentro de limites bem definidos (valor mínimo e valor máximo). Entretanto, a garantia de Qualidade de Serviço em redes de computadores envolve vários níveis de atuação em diversos tipos de equipamentos e tecnologias, ou seja, esses parâmetros não estão localizados em apenas um único equipamento ou componente da rede. Considerando esse fato, a Qualidade de Serviço deve atuar em todos os equipamentos, camadas de protocolo e entidades envolvidos.”[2]

QoS traz o comprometimento dos recursos que há dentro desse ambiente para manter a funcionalidade das aplicações e a integridade dos dados. Na visão técnica dentro das redes tradicionais a qualidade de serviço fica ligada ao conceito de diferenciação do tratamento dado aos pacotes e a minimização do atraso do envio.

Sérgio Souza³ deixa bem definido o que são as características de roteamento na qualidade de serviço:

“Manter as informações sobre o estado da rede o mais atualizada possível, pois disto depende a escolha das rotas – mas por outro lado, recursos com

³ http://mum.mikrotik.com/presentations/BR08/Aplicando_politicas_de_QoS.pdf

informações de controle sobre o estado dos enlaces devem ser minimizados.

O roteamento com QoS deve ser realizado de maneira eficiente, atendendo às exigências de qualidade demandada, sem contudo, consumir muito tempo de processamento. Então, o tempo de resposta deve ser baixo, sob pena de não atender à algum requisito da aplicação e surgirem problemas: pacotes descartados (dropped packets); atraso (delay); entrega desordenada (out of order); erros.”

Há vários parâmetros que podem ser negociados em uma rede de dados:

- *Throughput* (vazão)
- *Delay* (atraso na transmissão)
 - *Atraso de enfileiramento nos roteadores
 - *Atraso de codificação/decodificação
- Taxa de perda
- *Jitter* (variação de atraso)
- Largura de banda

Uma arquitetura de QoS deve prover, basicamente, as seguintes funções:

- Definição de parâmetros;
- Mapeamento dos parâmetros definidos;
- Negociação e controle de admissão;
- Reserva de recursos;
- Monitoramento;
- Renegociação e correções.

“A QoS não é um hardware ou um software, ou algo específico, mas uma arquitetura, um conjunto, uma combinação variada de soluções fim-a-fim, e, justamente por isso, necessita de uma série de considerações novas e já tradicionais – iniciando-se, obrigatoriamente, em um criterioso planejamento.”[3]

Parâmetros de QoS:

APLICAÇÃO	VAZÃO (TÍPICA)
Aplicações Transacionais	1 Kbps a 50 Kbps
Voz	10 Kbps a 120 Kbps
Aplicações Web (WWW)	10 Kbps a 500 Kbps
Transferência de Arquivos (Grandes)	10 Kbps a 1 Mbps
Vídeo (StreamingII)	100 Kbps a 1 Mbps
Aplicação Conferência	500 Kbps a 1 Mbps
Vídeo MPEG	1 Mbps a 10 Mbps
Aplicação Imagens Médicas	10 Mbps a 100 Mbps
Aplicação Realidade Virtual	80 Mbps a 150 Mbps

Figura 1: Tabela de Vazão Típica de Aplicações em Rede
 Fonte: www.logicengenharia.com.br/mcamara/ALUNOS/QoS.pps

“De uma maneira geral, todos os parâmetros são importantes para garantir a Qualidade do Serviço, eles irão variar de acordo com a necessidade de cada aplicação, ou seja, alguns parâmetros terão mais importância que outros, a depender do enfoque do negócio.” [4]

1.1 COMO OBTER QoS

Dentro de um estudo relacionado a aplicações os parâmetros de Qualidade de Serviço e assim foram sugeridas propostas que foram desenvolvidas pelo IETF (*Internet Engineering Task Force*) para implementação da qualidade na internet como: MPLS; Serviços Integrados (*IntServ*); Serviços Diferenciados (*DiffServ*).

As duas variações mais conhecidas são:

- *IntServ*
- *DiffServ*

Com base no trabalho de Samuel e Saulo[4], podemos ver as seguintes aplicações da QoS:

- IntServ – Serviços Integrados

- Garantia “fim-a-fim”
- Pontos positivos
 - Maior garantia
 - Eficiência em redes de curta distancia
- Pontos negativos
 - Não é aconselhada para longas distâncias
 - Maior custo

- DiffServ – Serviços Diferenciados

- Bastante difundido
- Forma de operação
- Pontos positivos
 - Custo
 - Boa aplicação em redes de longa distância
 - Facilidade de implementação
 - Uso de classes de serviços
- Pontos negativos
 - Não possui total garantia

- MPLS – MultProtocol Label Switching

- Uso de cabeçalhos, similar ao DiffServ

- Mais voltado para tráfego da rede
- Diferenças entre MPLS, o IntServ e o DiffServ
 - Não é controlado pela aplicação
 - É residente apenas nos roteadores
 - É independente do protocolo de rede.

“IntServ: Introduzido pelo IETF em 1994 (RFC 1663), trata-se de uma sinalização de QoS, onde o transmissor comunica suas necessidades de Qualidade de Serviço para que a rede providencie todos os recursos necessários”.[3]

A idéia geral de *IntServ* é que qualquer roteador no sistema possa suportar sua tecnologia e que qualquer aplicação exija um nível de garantia sendo assim responsável por fazer suas tarefas individualmente.

“DiffServ: uma proposta de arquitetura (RFC 2475) contrária à orientação por fluxo do RSVP, implementando diferenciação de serviços de forma escalonável na internet. Nessa arquitetura existe um esquema de classificação e marcação de pacote ao longo do caminho e operações mais sofisticadas somente são realizadas nos limites da rede (nos hosts).”[3]

Como para *DiffServ* que opera sobre grande volume de dados, isso pode acabar implicando na negociação do recebimento de pacotes. Os acordos que se resultam nesse parâmetro são designados “acordos de nível de serviço”(*Service Level Agreements*). Estes especificam que classe de tráfego serão servidas e quais as garantias da classe e seu volume.

Qualidade de serviço vem então ser o tratamento diferenciado do tráfego reunido em classes de serviço, com o objetivo de garantir o nível de serviço adequado a cada aplicação.

Para que uma aplicação tenha desempenho consistente e previsível é necessário observar os requisitos de *throughput*, atraso, *jitter* e perda de pacotes da mesma. Podemos visualizar melhor o conceito na figura a seguir e assim poder concluir que sempre será necessário implementar a qualidade de serviço fim-a-fim para obter o resultado esperado.



Figura 2: Aplicação de QoS

Fonte: <http://imasters.com.br/artigo/13011/redes-e-servidores/conceitos-basicos-de-qos>

Para concluir este capítulo é interessante ressaltar a conclusão de Douglas Falsarella⁴,

“A qualidade de serviço deverá ser implementada fim-a-fim em todos os pontos da rede, pois se um determinado trecho não suportar e respeitar as definições da política de QoS, passa a ser "o ponto de falha". A qualidade de serviço da rede será dada pela qualidade de serviço de todos os elementos da rede.

O roteamento pode ser considerado uma forma primitiva de QoS. Ele consiste no encaminhamento de pacotes para o seu destino utilizando a melhor rota possível. A mesma preocupação também tiveram os desenvolvedores do projeto original do IP (RFC 791), quando foi reservado um campo no cabeçalho IP chamado Type of Service (ToS), para oferecer diversos perfis de serviços. Da mesma forma, o protocolo TCP foi implementado com algoritmos de controle de congestionamento que foram aprimorados ao longo do tempo.

Todas estas iniciativas de implementar serviços diferenciados demonstram que a preocupação com QoS é antiga.” [5]

⁴ **Douglas Falsarella** atualmente é especialista em arquitetura de redes da empresa todo. É formado em Ciência da Computação e especialista em Routing & Switching, bem como em projetos de Rede e Telecom. Atuou em empresas como Huawei, Alcatel-Lucent, AT&T, CTBC, Telefônica Internacional. -

2. ESCALONAMENTO DE PROCESSOS

O conceito de processo é importante no estudo de sistemas operacionais. Entende-se melhor sobre este conceito quando se pensa em um computador funcionando em multiprogramação, ou seja, o uso de vários programas que estão na memória do computador.

“O conceito básico que gerou a implementação foi à possibilidade de a UCP ser compartilhada entre diversos processos. Para isso todo sistema multiprogramável possui critério para determinar qual a ordem na escolha dos processos para execução, dentre os vários que concorrem pela utilização do processador. Esse procedimento de seleção é uma das principais funções realizadas por um sistema operacional, e é conhecido como escalonamento (*scheduling*). Os principais objetivos do escalonamento são basicamente, manter a UCP ocupada a maior parte do tempo, balancear a utilização do processador entre diversos processos, maximizar o *throughput*⁵ do sistema e oferecer tempos de resposta razoáveis para usuários interativos. Esses objetivos devem ser atendidos de forma que o sistema trate todos os processos igualmente, evitando, assim, que um processo fique indefinidamente esperando pela utilização do processador (*starvation*).” [6].

Quando um processo usa a UCP, o mesmo está no estado executando uma tarefa, e se está esperando o término de um serviço está em estado bloqueado. Mas se o processo fica pronto para execução e não é executado o mesmo está pronto.

“O sistema operacional mantém uma lista (fila) dos processos que estão prontos, a chamada lista de processos prontos (*ready list ou ready queue*). O

⁵ Taxa de Transferência

diagrama da figura abaixo mostra como os estados de um processo podem mudar durante a execução.”⁶

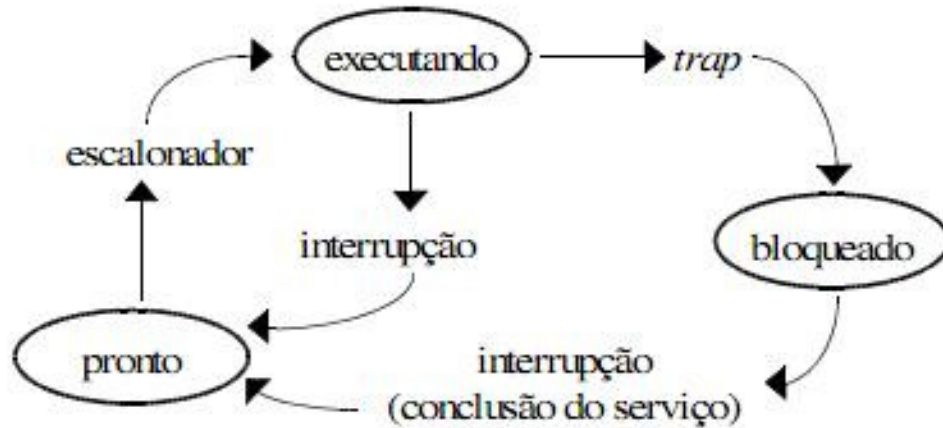


Figura 3: Estados sucessivos de um processo no sistema

Fonte: http://www.facom.ufu.br/~faina/BCC_Crs/INF10-2S2005/DownLoad/ch5-p1-aline-marcelo.pdf

Baseado no estudo de Aline e Marcelo[7], temos uma breve apresentação sobre escalonamento de processos:

- Como memórias e terminais, a CPU é um recurso dividido para cada processo no sistema.
- O escalonador é o componente do sistema operacional que determina qual processo vai rodar num dado tempo e quanto tempo vai rodar.
- Sistema de partilha de tempo – permite que diversos processos funcionem simultaneamente. Em um sistema monoprocessado cria uma ilusão de execução simultânea, intercalando processos em uma base da partilha de tempo, antes de trocar

⁶ Fonte: Escalonamento de Processos: http://www.facom.ufu.br/~faina/BCC_Crs/INF10-2S2005/DownLoad/ch5-p1-aline-marcelo.pdf

por outro processo. Este período é chamado *quantum* do tempo ou fatia do tempo.

- Dois aspectos do escalonador: Política – A regra usada para decidir qual processo vai rodar e quando ele será trocado por outro. Implementação – As estruturas de dados e algoritmos usados para cumprir tais políticas.

Para concluir o estudo acima citado, ainda temos os objetivos da política de escalonamento:

- Tempo de resposta rápido para aplicações interativas
- *Throughput* elevado para trabalhos rodando em *background*
- Execução da política eficientemente e com despesas gerais e mínimas.
- Mudança de contexto – o escalonador arranja para o processador trocar de um processo para o outro.
- O *Kernel* salva o contexto da execução do processo atual em seu bloco de controle do processo (PWB) – Parte da área de um processo
- O contexto é uma cópia dos valores do propósito geral, gerenciamento de memória e outros registradores especiais do processo. O *kernel* carrega os registradores com o contexto do processo seguinte a ser rodado, obtido do PWB do processo seguinte. A CPU começa a executar o processo seguinte.
- Mudanças de contexto – Uma operação cara. Além de salvar uma cópia dos registradores do processo o *kernel* deve executar outras tarefas do específico da arquitetura – limpar os dados, instruções, ou traduções de endereços de cache para evitar acessos incorretos à memória pelo novo processo. Nas

arquiteturas *pipelined* (por exemplo, RISC), o *kernel* deve limpar as instruções de *pipelined*.

- Prévias para mudar o contexto – Estes fatores podem influenciar não só a implementação, mas também a política de escalonamento.
- O *clock* é crítico na operação de escalonamento, porque o escalonador geralmente quer preemptar processos executando quando sua fatia de tempo expira.

O escalonamento do processador é ligado a uma estratégia de técnicas de atribuição da CPU nos processos. Sendo assim, a base dos sistemas com multiprogramação.

“A execução de um processo consiste em geral de um ciclo de execução no CPU (*CPU burst*), seguido de uma espera por uma operação de I/O (*I/O burst*)”. [8].

Para explicar o que é *Burst* temos a seguinte definição:

“*Burst* é um recurso que permite satisfazer os requisitos de largura de banda adicional para o cliente mantendo a velocidade maior por um período limitado do tempo. O *Burst* é ativado quando o consumo máximo da conexão atingir duas horas de *download* contínuo o servidor identificará automaticamente a conexão e limitará para a velocidade designada no plano do contratante. O *Burst* é desativado após o termino da aplicação que ativar o *burst* o limite de *download* real será retornado e o servidor passará novamente a velocidade máxima inicial do plano”⁷.

⁷ Turbo link - Fonte de pesquisa: http://www.turbolink.com.br/portal/index.php?option=com_content&view=article&id=51:burst&catid=35&Itemid=41. Arquivo acessado em julho de 2011.

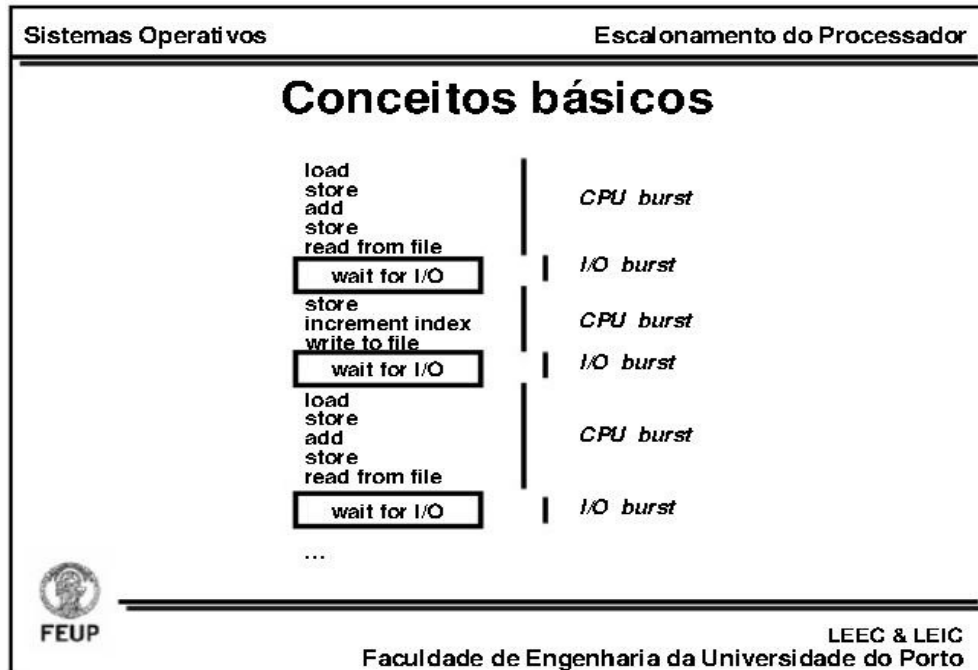


Figura 4: Conceitos básicos de um Escalonamento do Processador
 Fonte: <http://pt.scribd.com/doc/7202387/Escalonamento-de-Processos>

Para explicar o exemplo da figura anterior, segue figura explicativa:

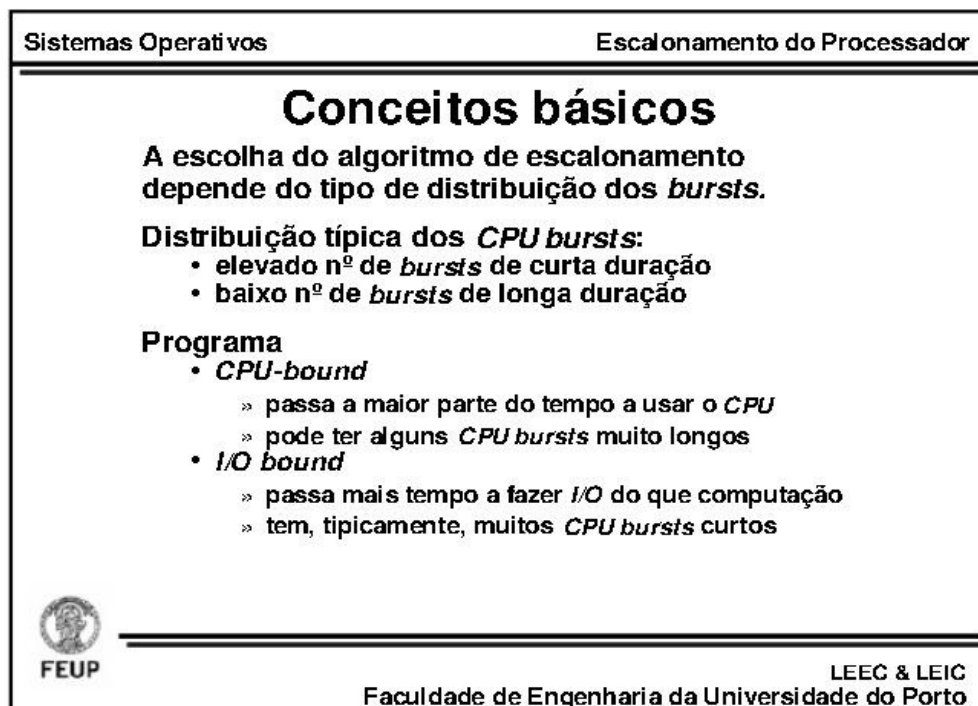


Figura 5: Explicação para o programa da figura anterior
 Fonte: <http://pt.scribd.com/doc/7202387/Escalonamento-de-Processos>

2.1 Níveis de Escalonamento

A política para o escalonamento é a estratégia do sistema que pode escolher os processos a serem executados a cada instante. Esta deve manter os critérios de maximizar o número de processos/tempo e minimizar o tempo que cada processo espera para ser executado para assim conseqüentemente evitar o adiamento indefinido.

Temos os seguintes níveis que compõe a política de escalonamento de uma CPU:

Escalonamento de longo prazo: “Determina que processos sejam admitidos para execução no sistema.” [8].

LP é responsável pelo grau de multiprocessamento fazendo assim a seleção de processos novos que são limitados por I/O sendo priorizados aqueles limitados.

Escalonamento de médio prazo: “Determina que processos são carregados total ou parcialmente, em memória principal, depois de terem estado suspensos. Está ligado a função *swapping*.”[8].

O processo utilizado para MP é envolvido com a memória secundária (*swap*) fazendo assim operações de *swapping in* e *swapping out*.

Escalonamento de curto prazo: “Determina qual processo a ser executado proveniente da fila de processos prontos.” [8].

Este pode fazer decisões de escalonamento de uso freqüente, em comparação com o escalonamento de médio e longo prazo.

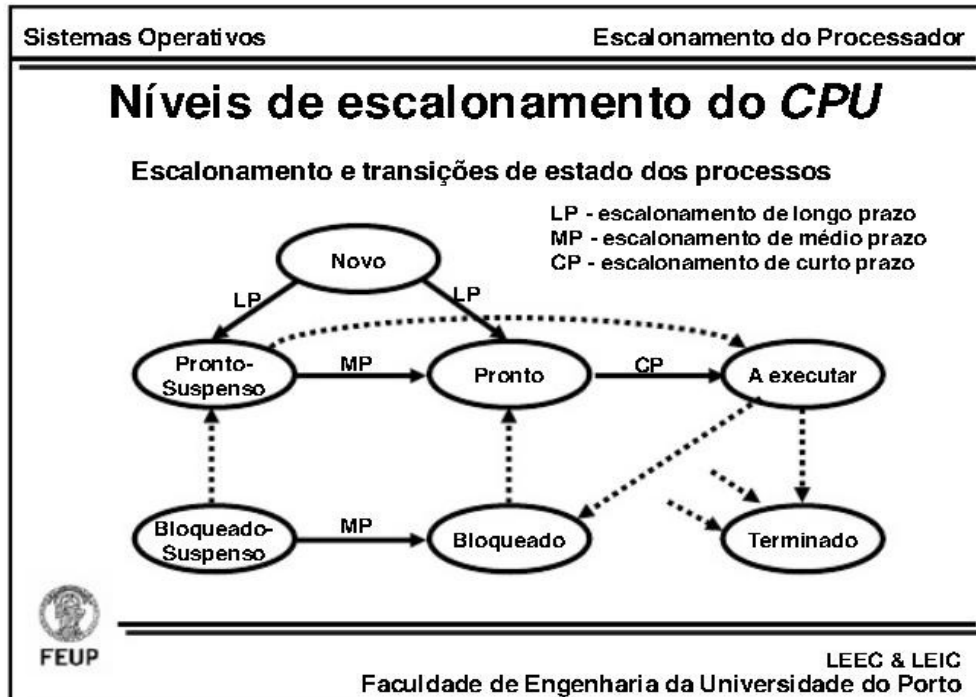


Figura 6: Escalonamento e transições de estado e processos
 Fonte: <http://pt.scribd.com/doc/7202387/Escalonamento-de-Processos>

De acordo com o estudo realizado pela FEUP [8], temos a seguinte descrição:

“Escalonamento de longo prazo:

- Intervém na criação de novos processos
- A decisão é geralmente apenas em função de:
 - ◆ Os recursos necessários e disponíveis
 - ◆ O número máximo de processos admissíveis
- Determina o grau de multiprogramação
 - ◆ Grau de multiprogramação = Número de processos em memória

Escalonamento de médio prazo

- Intervém por ocasião da escassez de recursos
- Pode ser executado com intervalos de alguns segundos a minutos

Escalonamento de curto prazo

- As decisões relativas ao escalonamento podem ter lugar quando um processo
 1. Comuta de “a executar” → “bloqueado”
 2. Comuta de “a executar” → “pronto”
 3. Comuta de “bloqueado” → “pronto”
- Em geral é invocado com intervalos muito curtos (algumas centenas de milissegundos)
- Pode ser:
 - ◆ Preemptivo – o processo pode ser forçado a ceder a CPU
 - ◆ Não-Preemptivo – o processo executa até bloquear ou ceder a vez voluntariamente

O módulo despacho (*dispatcher*) dá o controle da CPU ao processo selecionado pelo módulo de escalonamento de curto prazo, isto envolve:

- Comutação de contexto
- Comutação para modo utilizador
- “Saltar para o endereço adequado do programa do utilizador.” [8].

O tempo que decorre entre a paragem de execução de um processo e o início de outro é designado por latência de despacho.

2.2 Escalonamentos Preemptivo e Não Preemptivo

Para os processos que podem ser interrompidos, ou seja, que já estão em execução, para que outros possam ser executados se dá o nome de sistema

preemptivo. Para esse tipo de escalonamento o sistema gera uma atenção prioritária, que seria um caso de processo executado em tempo real. O sistema preemptivo permite o compartilhamento maior e uniforme do processo.

Concluindo a idéia de escalonamento preemptivo também temos:

“Outro benefício, decorrente deste tipo de escalonamento é o compartilhamento do processador de uma maneira mais uniforme entre os processos. A troca de um processo por outro na UCP (mudança de contexto) causada pela preempção gera um overhead ao sistema. Para que isso não se torne crítico, o sistema deve estabelecer corretamente os critérios de preempção.” [6].

O escalonamento não-preemptivo é o mais simples, por ser um processo que recebe o controle da CPU e não perde o mesmo até que seja concluído o processamento. Sistemas não-preemptivos são conhecidos como cooperativos, pois exigem cooperação para que todas as tarefas possam ser executadas.

“Nos primeiros sistemas multiprogramáveis, onde predominava tipicamente o processamento batch⁸, o escalonamento implementado era do tipo não-preemptivo. Nesse tipo de escalonamento, quando um processo (ou job) ganha o direito de utilizar a UCP, nenhum outro processo pode lhe tirar esse recurso.” [6].

Para finalizar a idéia dos dois tipos de escalonamento, conclui-se que para o sistema não-preemptivo que o sistema pode conduzir um processo para a execução, mas ele não poderá tirar o mesmo de execução.

No caso do escalonamento preemptivo, o sistema será capaz de interromper a execução do processo e depois poderá reconduzir este novamente ao processador, mas em um novo momento.

⁸ No contexto de sistemas operacionais, é um termo referente a um processamento de dados que ocorre através de um lote de tarefas enfileiradas, de modo que o sistema operacional só processa a próxima tarefa após o término completo da tarefa anterior.

3. ALGORITMOS DE ESCALONAMENTO

Hoje em dia a rede de dados é muito grande, contam com muitos pacotes trafegando em milésimos de segundos, estes entram por um nó e saem por outro. Neste percurso, os pacotes são controlados por um mecanismo de controle de tráfego.

“Quando um computador é multiprogramado ele muitas vezes tem variados processos que competem pela CPU ao mesmo tempo. Essa situação ocorre sempre que dois ou mais processos estão simultaneamente no estado pronto. Se somente uma CPU se encontrar disponível, deverá ser feita uma escolha de qual processo executará em seguida. A parte do sistema operacional que faz a escolha é chamada escalonador, e o algoritmo que ele usa é o algoritmo de escalonamento.” [9].

Os algoritmos de escalonamento (*scheduling algorithms*) são componentes essenciais para atingir o objetivo de integração de serviços que geram QoS nas categorias de serviços, e assim podendo oferecer garantias de desempenho e aplicações críticas na rede de dados.

Um algoritmo de escalonamento executa duas funções: 1) Decide a ordem de serviço de fluxos em competição (pedido de serviço); 2) Gera filas de pedidos de serviço.

Acompanhando o estudo da Tese do Professor Kleber Nabas (2009), temos a seguinte definição de algoritmos de escalonamento:

“O compartilhamento de recursos em redes de comunicações fez com que os sistemas adotassem uma estratégia de multiplexação estatística, devido a falta de recursos ofertados pela rede. Isto pode ser percebido pela contenção de pacotes existente em roteadores da rede. Em redes que suportam integração de serviços torna-se necessário providenciar QoS diferenciada por categorias de serviço, oferecendo garantias de

desempenho a aplicações críticas e ao mesmo tempo permitindo uma partilha de recursos de acordo com critérios de equidade. Os algoritmos de escalonamento são componentes essenciais para atingir estes objetivos, uma vez que determinam as disciplinas de serviço a aplicar a fluxos aceitos pela rede. Os algoritmos de escalonamento são usados em qualquer sistema ou camada protocolar em que ocorra contenção por recursos, mas em especial em elementos de rede por exemplo em roteadores. Eles também desempenham um papel importante na provisão de diferentes níveis de QoS a diferentes aplicações, permitindo controle diferenciado de atraso, largura de banda ou taxa de perdas de pacotes. Em geral pretende-se que os algoritmos de escalonamento sejam simples (fáceis de implementar) e tratem de forma equitativa tráfego do tipo melhor esforço e garantam ao restante do tráfego níveis de desempenho negociados e facilitem os mecanismos associados de controle de admissão de conexões. Podem ser citados alguns escalonadores de pacotes: FIFOQ, FQ, CQ, GPS, WFQ, WRR e PQ.” [10].

Para que um escalonamento aconteça precisam ser tomadas várias decisões, pois há várias situações para se utilizar o escalonamento necessário e assim executar o processo.

“Uma decisão de escalonamento deve ser tomada ao término de um processo. Como processo não pode executar mais (já que ele não mais existe), algum processo deve ser escolhido entre processos prontos. Se nenhum processo estiver pronto, é executado um processo ocioso gerado pelo sistema. (Tanenbaum, 2003).

3.1 ESCALONAMENTO FIFOQ (*First In First Out Queueing*)

Este é um mecanismo de escalonamento, o mais simples entre os tipos de escalonadores.

“Este é o algoritmo de escalonamento de pacotes utilizado originalmente no conjunto de protocolos Inter Rede. O algoritmo está baseado na criação de um buffer de tamanho fixo para cada interface física. Os pacotes que devem ser enviados por esta interface são armazenados neste buffer e enviados em ordem de chegada – o primeiro pacote a ser transmitido será o pacote mais antigo no buffer.” [10].

Na figura abaixo podemos ver a chegada de pacotes, e quando não há mais espaço no *buffer*, os pacotes que vão chegando automaticamente vão sendo descartados.

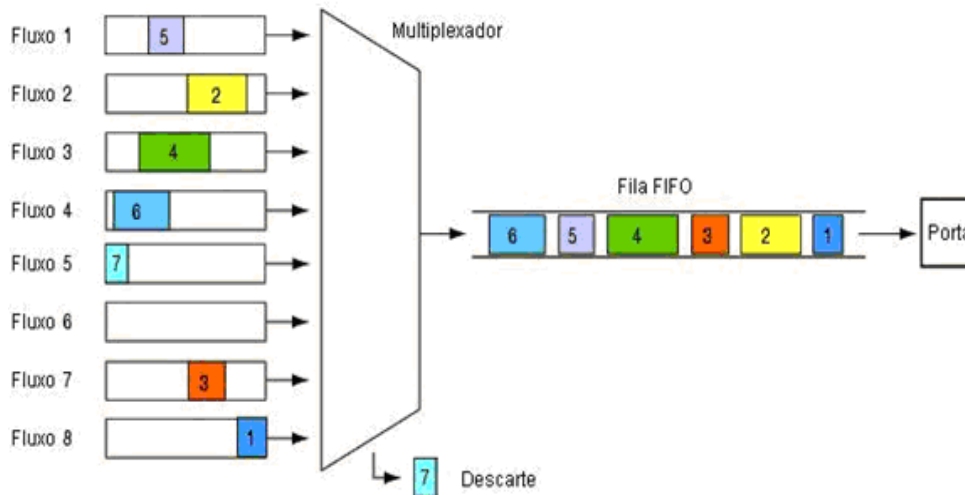


Figura 7: Escalonador FIFOQ (BALLIACHE, 2003)
 Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 4

“O problema deste método é a impossibilidade da separação do tráfego e a divisão da banda passante entre os fluxos IP.” (Kleber, 2009).

As vantagens de se usar o FIFOQ é que seu comportamento é previsível; este fornece um método de contenção simples para recursos da rede enquanto o tamanho da fila permanece curto.

Segundo o estudo sobre mecanismos de escalonamento podemos ver que há várias limitações para o uso deste escalonamento:

“Uma única fila FIFO não permite a organização em diferentes categorias. As aplicações que usam UDP são favorecidas relativamente às que usam TCP em períodos de congestão de rede. O impacto do atraso influência de forma idêntica todos os fluxos. Um fluxo “mal comportado” pode consumir todo o espaço de memória de uma fila de espera FIFO. As consequências destas limitações causam um aumento do atraso, do jitter (variação de atraso sofrido por um pacote) e perdas para todos os outros fluxos UDP e TCP. Quando a rede IP opera sem sobrecarga, as filas são necessárias somente para assegurar que os pacotes sejam descartados quando existe uma sucessão rápida, ininterrupta e de curta duração de tráfego de dados. Nessa situação, este mecanismo é altamente eficiente, na medida em que o tamanho da fila permanece pequeno. Entretanto, quando a carga da rede aumenta, o tráfego rápido causa um atraso significativo de colocação na fila em relação ao atraso da transmissão total e, quando a fila está totalmente cheia, todos os pacotes subsequentes são descartados. Se a fila opera

deste modo por longo período, a qualidade de serviço inevitavelmente se degradará. Portanto, pode-se apontar como principal deficiência deste algoritmo o fato deste usar a ordem de chegada de pacotes para determinar a prioridade destes.”[11].

Pode concluir-se então que o algoritmo FIFOQ seleciona os pacotes para transmissão na mesma ordem em que eles chegam, e assim podem ocasionar a perda de pacotes nas filas. Ocasionalmente assim uma baixa qualidade para um escalonador de pacotes.

3.2 ESCALONAMENTO FQ (*Fair Queueing*)

No algoritmo FQ as mensagens são ordenadas em sessões, e para cada sessão representa um fluxo, onde estes fluxos são atendidas por *round-robin*, um pacote por fluxo. No caso de uma sessão lotar seu *buffer*, os próximos pacotes são descartados, independentemente das demais sessões estarem completas, conforme é mostrado na Figura 8. Isto faz com que um fluxo não tenha uma parcela maior do meio em detrimento dos demais. (Kleber, 2009).

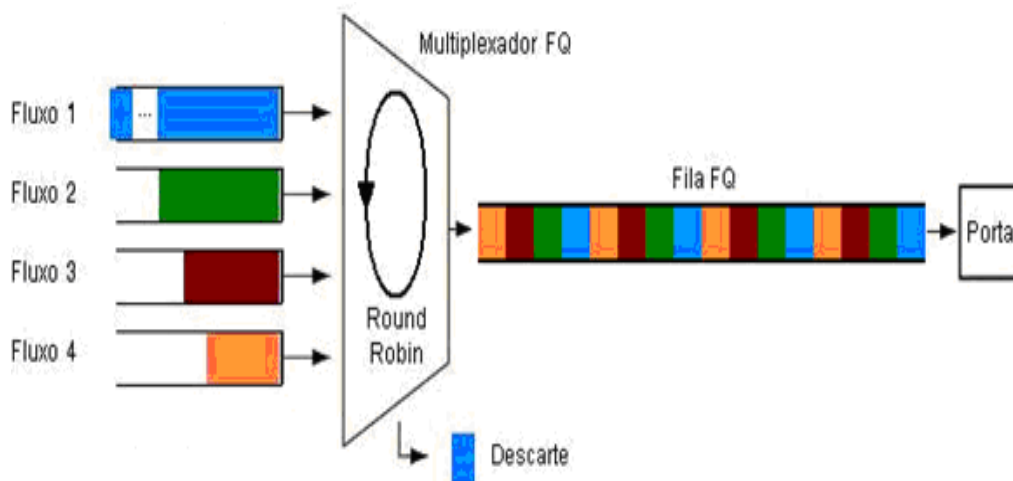


Figura 8: Escalonador FQ (BALLIACHE, 2003)
 Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 5

3.3 ESCALONAMENTO CQ (*Custom Queueing*)

O CQ aloca partes da banda para determinados fluxos e a outra parte é dividida para os demais fluxos. O CQ funciona de forma semelhante ao FQ, porém a cada tempo do *round-robin* são liberados os pacotes conforme a proporção alocada inicialmente para cada fluxo. (Kleber, 2009).

Nesta técnica, tem-se uma fila para cada tipo de tráfego especificado.

Na figura 9 temos um exemplo de implementação do algoritmo CQ, feito pela CISCO:

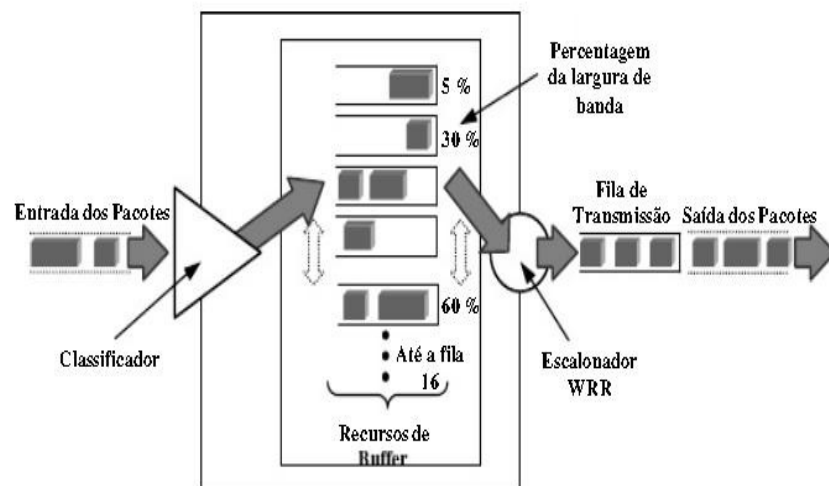


Figura 9: Custom Queueing

Fonte: <http://pt.scribd.com/doc/56415876/18/Weighted-Fair-Queueing-WFQ>. Consultada em Julho/2011.

3.4 ESCALONAMENTO GPS (*Generalized Processor Sharing*)

É o melhor escalonador possível, no entanto não é implementável. Durante qualquer intervalo de tempo em que existam N filas não vazias, o servidor serve simultaneamente os N primeiros pacotes de cada fila, cada um com um débito igual a $1/N$ da capacidade da ligação física. De forma equivalente se pode dizer que é servida uma quantidade de dados infinitesimal em cada visita a uma fila (uma

aproximação consistiria em servir rotativamente um bit de cada fila não vazia). (Kleber, 2009).

3.5 ESCALONAMENTO CBQ (*Class Based Queueing*)

Este algoritmo foi projetado para receber vários tipos de aplicações com especificações de largura de banda mínima e exigência de atraso que possa haver em uma rede. O seu buffer pode ser dividido em várias filas com diferentes níveis de classes.

“Este algoritmo permite definir a quantidade de bytes de dados que deve ser retirada de cada fila do buffer para a fila da interface de saída do dispositivo de rede. O CBQ é o mecanismo de escalonamento mais complexo existente atualmente. É constituído por dois escalonadores: um genérico e um de partilha de ligação. O escalonador genérico tenta garantir aos fluxos com requisitos de tempo real, baixos atrasos na fila de espera, enquanto que o de partilha de ligação impede que as classes de tráfego de tempo real monopolizem a utilização da ligação, e faz também a distribuição de debito excedente. Baseia-se numa estrutura hierárquica em arvores de classes de tráfego, sendo cada pacote classificado à entrada dos nós.” [11].

A figura 10 mostra a base de um escalonamento CBQ:

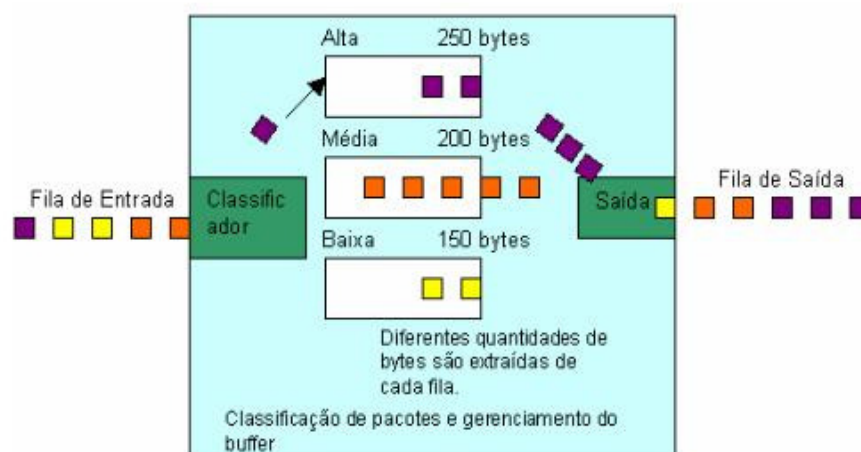


Figura 10: Escalonamento CBQ

Fonte: <http://www.slideshare.net/pedronero/quality-of-service-and-linux>

Para este algoritmo, um dos principais usos são os serviços diferenciados de redes complexas, com requisitos específicos para fluxos heterogêneos e em tempo real.

3.6 ESCALONAMENTO WFQ (*Weighted Fair Queueing*)

O mecanismo WFQ se inspira no algoritmo denominado BRR (Bit-by-bit-Round Robin), onde cada fluxo é mantido em uma fila de saída exclusiva e um bit de cada fluxo é enviado pelo enlace de saída de cada ciclo ali correspondido.

O WFQ é um algoritmo FQ melhorado onde é possível estabelecer pesos para determinados tipos de fluxo. O algoritmo escalona o tráfego prioritário para frente da fila, reduzindo o tempo de resposta. Ao mesmo tempo, compartilha o restante da banda com os outros tipos de fluxo de uma forma justa. (Kleber, 2009).

A seguir figura com a demonstração do escalonamento WFQ:

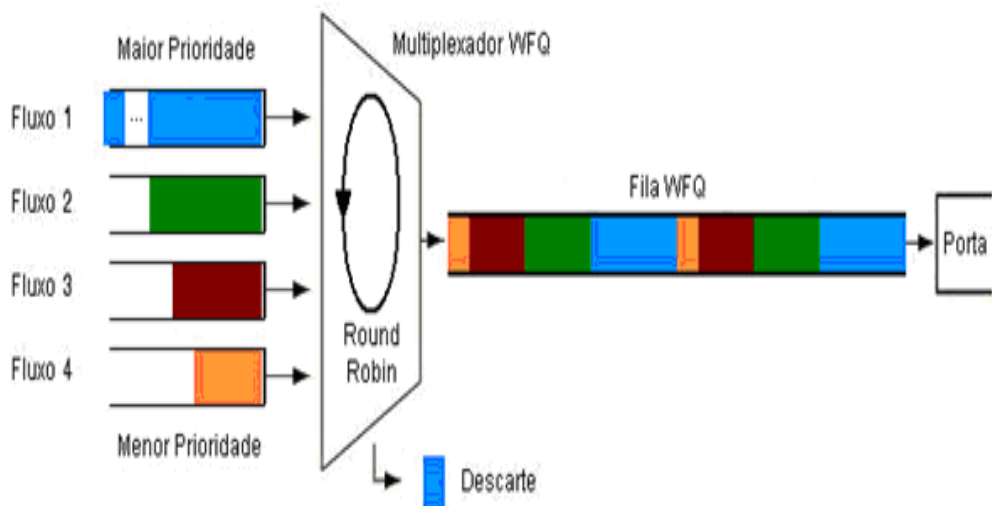


Figura 11: Escalonador WFQ (BALLIACHE, 2003).
Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 6

A figura 12 ilustra um sistema com duas filas, uma para tráfego de voz e outra para os fluxos de dados; para o tráfego de voz aloca-se um mínimo de 80Kbps, dos 128Kbps disponíveis no enlace.

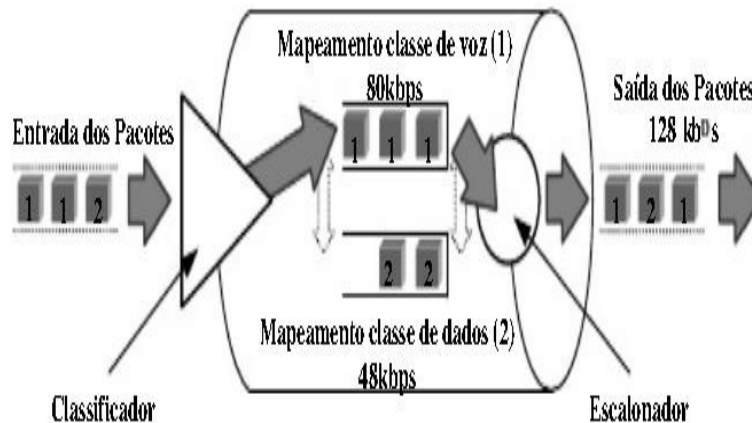


Figura 12: Exemplo de compartilhamento de tráfego de voz e dados utilizando WFQ
 Fonte: <http://pt.scribd.com/doc/56415876/18/Weighted-Fair-Queueing-WFQ>. Consultada em Julho/2011.

“O WFQ é dinâmico e se adapta automaticamente as mudanças das condições de tráfego” (Kleber, 2009).

3.7 ESCALONAMENTO WRR (*Weighted Round Robin*)

Cada fluxo irá fazer uso de uma fila no elemento de rede. O escalonador seleciona pacotes das filas em uma seqüência circular entre as filas. A quantidade de pacotes selecionados será igual ao peso da fila. Esse escalonador tem uma aproximação razoável com o GPS se os pacotes tiverem comprimento fixo e se as filas tiverem o mesmo peso, não é justo se os pacotes tiverem diferentes comprimentos e os pesos não forem iguais. (Kleber, 2009).

3.8 ESCALONAMENTO PQ (*Priority Queueing*)

O PQ é um algoritmo semelhante ao WFQ, porém os níveis mais prioritários têm total preferência em relação aos de menor prioridade. Contudo esse tipo de algoritmo deve ser muito bem administrado, pois quando houver um grande fluxo de pacotes de alta prioridade, os pacotes de baixas prioridades podem sofrer um atraso muito grande ou até mesmo o descarte. (Kleber, 2009).

3.9 ESCALONAMENTO HPF (*Highest Priority First*)

Este é um escalonamento por prioridades, é um algoritmo variante do escalonamento FIFO.

“Os processos em espera pelo processador são organizados numa fila segundo sua prioridade, sendo colocados a frente os processos *jobs* de maior prioridade, isto é, sua função de seleção é *max* (prioridade), favorecendo os processos considerados mais importantes.” [12]

Este é uma forma de escalonamento não preemptivo, segue representação do mesmo na próxima figura.

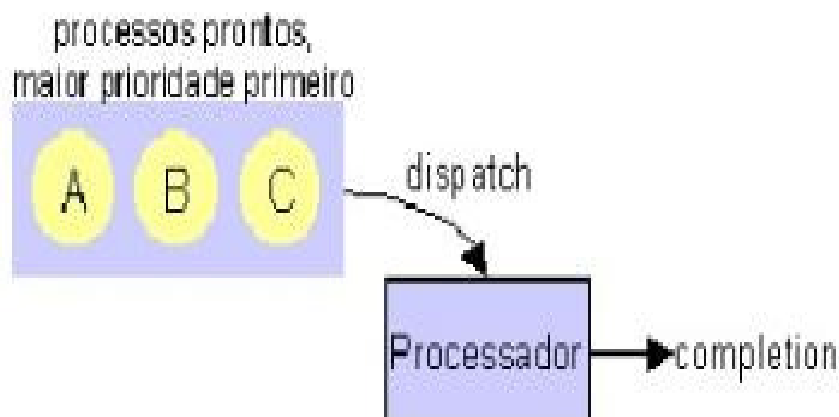


Figura 13: Escalonamento HPF

Fonte: <http://pt.scribd.com/doc/56415876/18/Weighted-Fair-Queueing-WFQ>. Consultada em Julho/2011

Com base no estudo sobre qualidade de escalonamento de pacotes, como referência vamos considerar a tese do Professor Kleber Nabas para uma demonstração mais ampla sobre o escalonador WFQ. Este demonstra qualidades superiores em desempenho e compartilhamento de banda.

4. O ESCALONADOR WFQ

O escalonamento WFQ (Enfileiramento Justo Ponderado) é um pacote de dados que permite prioridades de agendamento. É também denominado PGPS (*Packet Generalized Processor Sharing*).

É um escalonador de pacotes que tem se mostrado bastante popular, possuindo grande aceitação na indústria e servido como base para vários outros trabalhos. Para compreendê-lo, faz-se necessário explicar o modelo teórico do escalonador GPS⁹ ótimo, o qual o WFQ tenta aproximar. (Kleber, 2009).

O WFQ suporta até oito horas de filas de transmissão. Ele é um algoritmo de enfileiramento baseado em fluxo de tráfego nos horários de baixo volume em primeiro lugar. Para o alto volume de tráfego vai utilizar a largura da banda restante, ou seja, ele nunca desperdiça largura de banda; Se não há um tráfego de prioridade maior, o pacote de prioridade mais baixa é encaminhado na rede. Sendo assim para cada fila há uma percentagem da capacidade do enlace que é proporcional ao peso que é atribuído ao mesmo.

“Em 1993, Parekh e Gallanger (PAREKH; GALLANGER, 1993) demonstraram que, empregando servidores GPS nos comutadores da rede, garantias de atraso fim-a-fim poderiam ser providas para uma conexão cujo tráfego tivesse sido policiado por um mecanismo Leaky Bucket (GIROUX; GANTI, 1998). O algoritmo apresentado por Parekh e Gallanger apresenta um desempenho aproximado para o GPS, uma vez que o GPS assume que o servidor pode atender várias conexões simultaneamente e que o tráfego dessas conexões pode ser dividido infinitamente (não é realizável na

⁹ O objetivo deste algoritmo é a atribuição de valores de acordo com o critério *max-min fairness*, é também uma disciplina *work-conserving* (não implementável).

prática). Essa disciplina foi chamada de Packet Generalized Processor Sharing (PGPS), e demonstrou-se idêntica à disciplina WFQ proposta em 1989 por Demers, Keshav e Shenker (DEMERS; KHESAV; SHENKER, 1990) (LU; ROBOTHAM, 2004). O GPS baseia-se no critério de compartilhamento por igual max-min, com uma modificação a qual atribui pesos que refletem uma diferenciação entre requisições (weighted fair share). Considere-se o exemplo geral de n requisições de tamanhos $s_1, s_2, s_3, \dots, s_i, \dots, s_n$ e pesos relativos associados $w_1, w_2, w_3, \dots, w_i, \dots, w_n$. Seja W a

$$W = \sum_i^n w_i$$

soma dos pesos relativos (). Considere ainda que a quantidade de recursos disponíveis seja igual a R . Inicialmente, o servidor tentará alocar ($w_1 * R/W$) recursos à primeira requisição. Se essa quantidade de recursos é maior que s_1 , então a demanda é completamente satisfeita e o balanço de recursos faltando será ($R-s_1$). Se a quantidade for menor, então a demanda será atendida apenas parcialmente e o balanço dos recursos remanescentes será ($R- (w_1 * R/W)$). Em ambos os casos há agora uma nova quantidade de recursos disponíveis a serem compartilhados entre as ($n-1$) requisições remanescentes, e o processo é aplicado novamente de forma circular (BLAKE et al., 1998). O GPS pressupõe um modelo fluido onde o tráfego é infinitamente divisível e o escalonamento é feito com granularidade ínfima entre os fluxos ativos a cada instante. Na realidade a implementação do modelo fluido não é viável, de modo que sua principal utilidade é como referência. O GPS é especialmente útil neste sentido porque possui características ótimas de isolamento entre classes de serviço, justiça e garantias de latência máxima. "[10].

Ainda complementando o estudo da tese acima referenciado, temos:

O escalonador de pacotes *Packet Generalized Processor Sharing* também conhecido como WFQ utiliza um algoritmo que procura emular o GPS com uma granularidade ao nível de pacotes. A implementação dessa emulação é realizada a partir de uma função utilizada para calcular o instante em que cada pacote seria transmitido caso o GPS estivesse implementado. Essa função, conhecida como função de tempo virtual, atribui a cada pacote um instante de tempo de início e um instante de tempo de fim, os quais correspondem aos tempos no sistema de referência GPS. A ordem em que os pacotes são transmitidos corresponde à ordem crescente dos instantes de tempo virtuais de fim de cada um deles (BLAKE et al., 1998). (Kleber, 2009).

A Figura 14 representa o algoritmo WFQ:

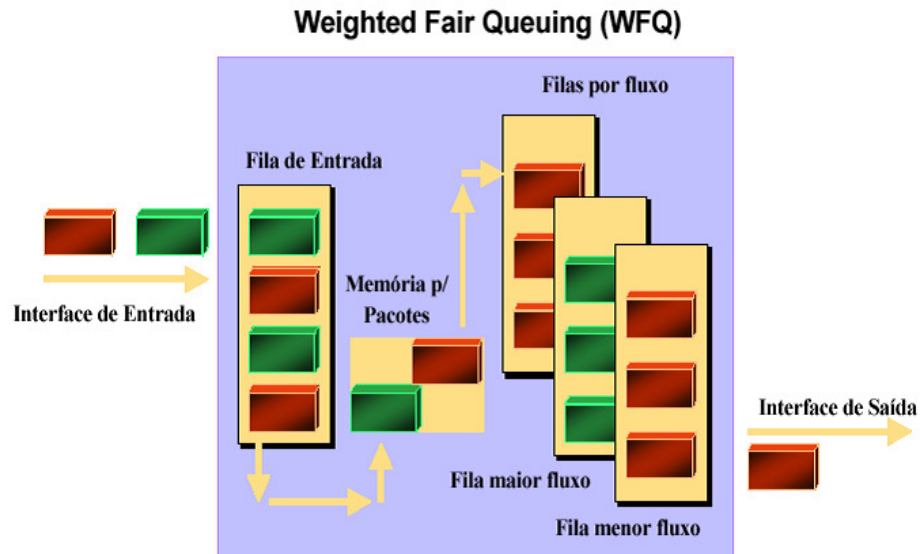


Figura 14: Escalonador de Pacotes WFQ (MELO, 2001)
 Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 8

A prioridade para os tráfegos deste algoritmo são determinadas pelos cálculos que levam em consideração “a taxa de bits da porta de saída, o número de filas ativas, a porção da largura da banda alocada para cada fila e o tamanho dos pacotes de cada fila, feito o cálculo, é atribuído um número para cada pacote e, desta forma, é definida a ordem de atendimento dos mesmos”. (Kulczyk, 2006).

A seguir outra figura mostrando o processo do escalonador WFQ.

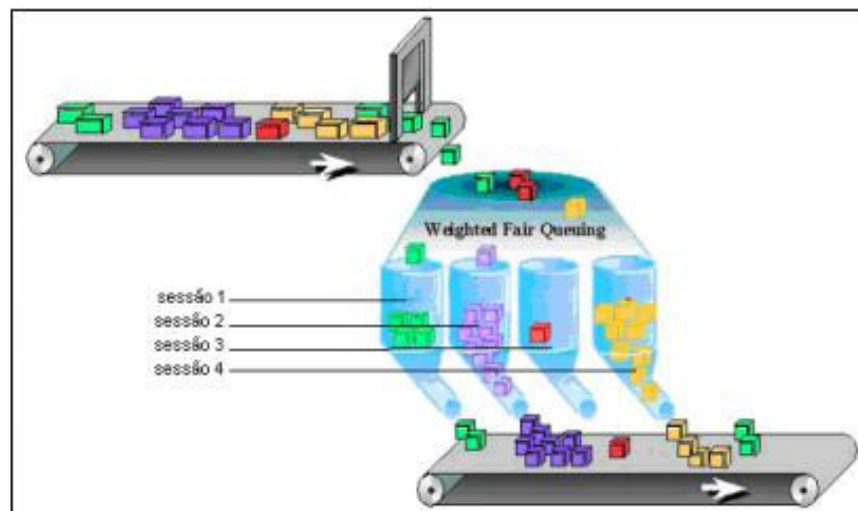


Figura 15: Exemplo de funcionamento do WFQ
 Fonte: http://www.dt.fee.unicamp.br/~renatoms/docs/monografia_renatomoraessilva.pdf

Os pacotes que chegam são classificados e enfileirados por sessões e o escalonador WFQ atende essas sessões de modo cíclico. Logo, ele atende primeiro a sessão 1, depois a sessão 2, a sessão 3 e, por fim, a sessão 4 (KUROSE E ROSS, 2003). Isso pode ser observado pelo exemplo de pseudocódigo do escalonamento WFQ, apresentado no quadro 1. (Silva, 2009).

```

enquanto houver pacotes faça
se há pacotes na fila então
    ciclo = ciclo + 1;
    saldo = saldo + cota;
    enquanto (saldo - tamanho_pacote) > 0
        saldo = saldo - tamanho_pacote;
        Transmite o pacote;
        i = i + 1;
        se fila vazia estiver vazia então
            saldo = 0;
            break;
    fim-se
fim-enquanto
fim-se
fim-enquanto

```

Figura 16: Quadro 1 – Pseudocódigo ao algoritmo de escalonamento WFQ
 Fonte: http://www.dt.fee.unicamp.br/~renatoms/docs/monografia_renatomoraessilva.pdf

No bloco de instruções apresentado na Figura 14, o tamanho do pacote é observado para verificar se o saldo disponível é suficiente para transmiti-lo. Se for suficiente, ele transmitirá o pacote e se não for suficiente, ele acumula o saldo para o próximo ciclo do escalonador, até que haja o valor suficiente para transmitir o pacote. No entanto, se a fila terminar antes do saldo ser esgotado, então o valor do saldo será zerado e um novo ciclo será iniciado para o tratamento de uma nova fila (JUNIOR, 2007). [14]

Neste algoritmo podemos encontrar muitas características interessantes de justiça e isolamento entre classes de serviços.

“Como no GPS, o WFQ procura implementar a justiça entre as classes de serviço utilizando o compartilhamento por igual max-min ponderado: se duas classes de serviço possuem pacotes enfileirados durante qualquer período de tempo, elas são servidas em proporção direta aos seus pesos, independentemente da quantidade de tráfego em excesso que qualquer um

deles possa ter gerado. Quando uma classe de serviço requer uma quantidade menor de recursos que o efetivamente alocado, o excesso é compartilhado entre todas as outras classes em proporção direta aos pesos relativos de serviço. O compartilhamento por igual max-min também implementa o isolamento entre as diversas classes de serviço: o tráfego em excesso de uma das classes não prejudica o tratamento ofertado ao tráfego das restantes. Deve-se frisar que o WFQ é apenas uma aproximação do GPS: enquanto o WFQ manipula pacotes de tamanho variado, o GPS pressupõe que o tráfego é infinitamente divisível. A aproximação implementada pelo WFQ é proporcional ao tamanho médio dos pacotes do tráfego servido. Mesmo assim, o WFQ pode ser considerado como uma grande melhora em precisão na justiça entre as classes de serviço e garantias de desempenho sobre outras disciplinas de fila como o Weighted Round Robin (WRR) (BLAKE et al., 1998). Uma outra vantagem do WFQ é que um limite do atraso fim-a-fim pode ser computado baseado no peso atribuído a cada fluxo. Esse limite pode ser calculado para todos os enlaces entre a origem e o destino do fluxo e é independente do número de fluxos que são multiplexados em cada enlace (REZENDE, 1999). Essa propriedade faz com que este algoritmo seja bastante utilizado em implementações de arquiteturas de qualidade de serviço que requerem limites quantitativos de atraso.” [10].

Para servir de exemplo, vamos utilizar a figura 16 que mostra um sistema simples com duas fontes compartilhando uma mesma fila e um único serviço.

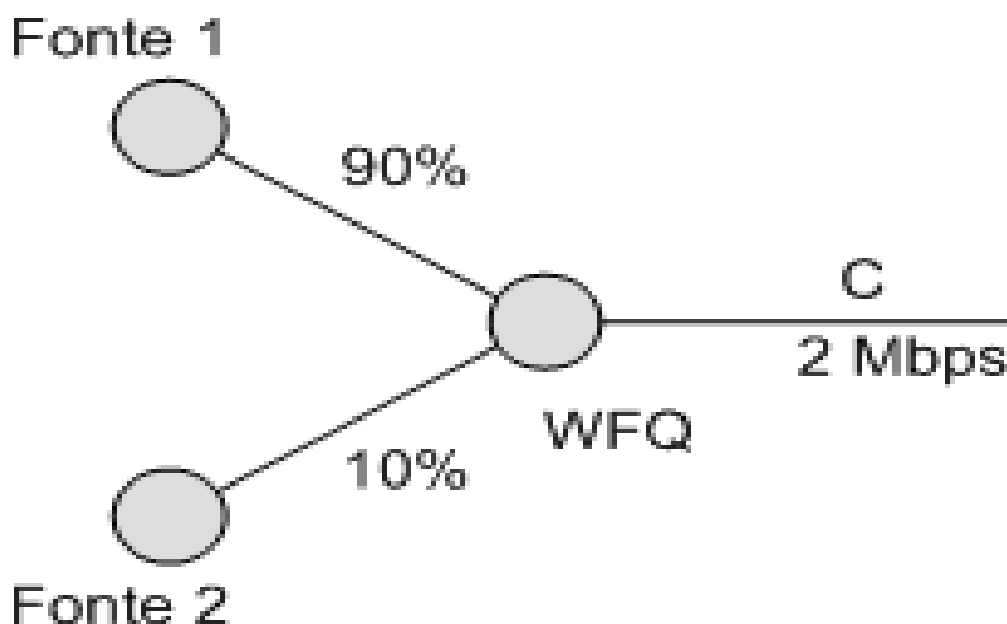


Figura 17: Configuração do Escalonador WFQ, compartilhando duas fontes com 90% e 10%.
Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 9

Através da simulação mostrada na figura acima podemos ver o gráfico na próxima figura que mostra as bandas ocupadas por cada um dos fluxos.

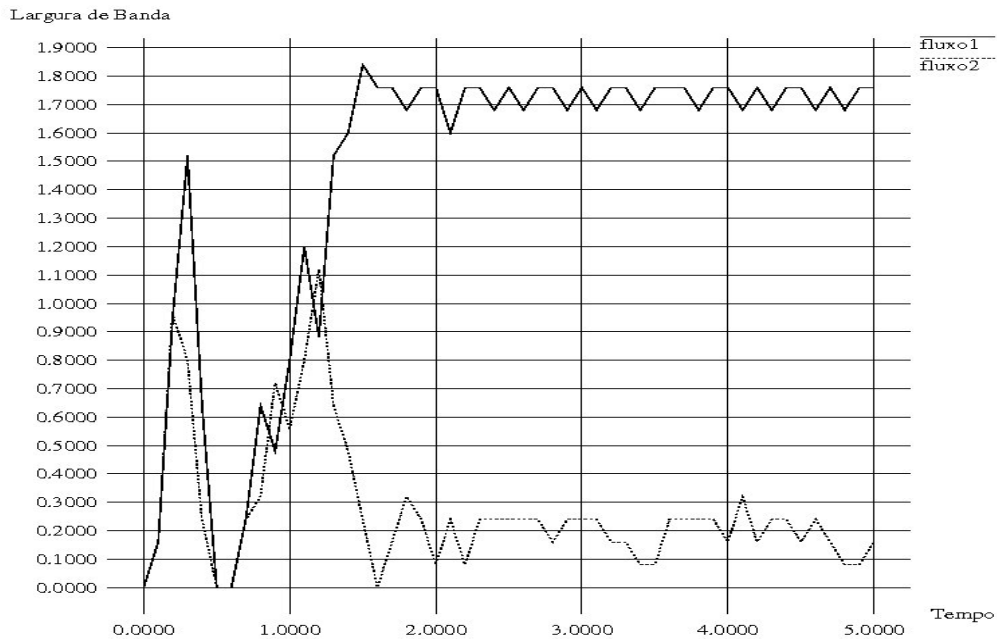


Figura 18: Largura de Banda utilizada pelos fluxos 1 e 2 da Figura 16
 Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 10

O enlace possui capacidade igual a 2Mbps e o WFQ está configurado com 90% para o fluxo1 e 10% para o fluxo2. Quando umas das fontes não têm dados a transmitir a outra fonte ocupa toda a banda disponível do sistema, isto é, o escalonador irá ponderar o peso das filas se e somente se as fontes tiverem dados para transmitir. Pode-se observar que a largura de banda é dividida proporcionalmente aos pesos do escalonador WFQ, isto é, 90% de 2Mbps é igual a 1,8Mbps e 10% é igual a 0,2Mbps. (Kleber, 2009).

4.1 ASPECTOS ANALÍTICOS DO WFQ

Dando continuidade ao estudo sobre o escalonador WFQ vamos analisar os aspectos analíticos do mesmo.

Com base na tese de Kleber Nabas temos a seguinte premissa: O escalonador WFQ pode gerenciar dois ou mais caminhos de dados e automaticamente suavizar rajadas e reduzir o tempo de atraso médio. O WFQ classifica pacotes por ordem de chegada e prioridade para determinar a ordem de transmissão. O efeito é que o WFQ classifica sessões em maior ou menor largura de

banda. Clientes aos quais é atribuída menor largura de banda são penalizados quando concorrem com outros que tem maior largura de banda (STEPHENS; BENNET; ZHANG, 1999). [10].

Vamos acompanhar a seguir a figura que demonstra o modelo de um nó de rede com o escalonador WFQ.

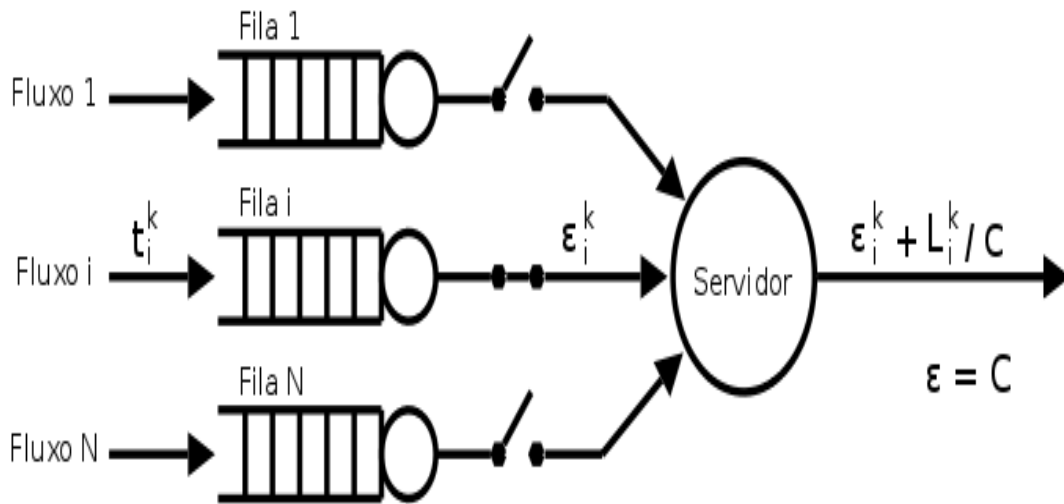


Figura 19: Modelo de um nó com o escalonador WFQ (DEKERIS; ADOMKUS; BUDNIKAS, 2006)
Fonte: Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 11

Para concluir o estudo do Prof. Kleber [10], ainda temos a definição da figura 18.

Este modelo pode ser usado para descrever o atraso nas filas, aonde t_i^k é o tempo de chegada do último bit de um pacote p_i^k (i.e., o k -ésimo pacote do i -ésimo fluxo) da fila. L_i^k representa o tamanho do pacote p_i^k , ϵ_i^k é o instante em que ocorre a transmissão do último bit do pacote p_i^k e C é a capacidade do enlace. O tempo gasto pelo pacote p_i^k na fila pode, então, ser escrito como:

$$T_i^k = \epsilon_i^k - t_i^k \quad (3.1)$$

O atraso de pacotes pertencente ao fluxo i pode ser descrito pela função:

$$\tau_i(t) = \sum_{k=0}^{\infty} L_i^k \cdot \delta(t - \epsilon_i^k) \quad (3.2)$$

Onde $\delta(t)$ é o delta de Dirac. Então o tempo de espera na fila em um nó de rede é:

$$T_i(s; t] = \int_s^t w_i(\tau) . d\tau \quad (3.3)$$

A base do escalonador WFQ é o cálculo dos pesos das filas, os quais influenciam diretamente o tempo real necessário para cada pacote ser servido por um sistema GPS ideal.

Tem-se λ_i como a taxa de chegada e ρ_i como a intensidade de tráfego (fator de utilização) para uma dada classe de tráfego i . Para simplificar os cálculos, assume-se que o tempo de serviço (razão entre λ_i e ρ_i) é igual a 1; então, $\lambda_i = \rho_i$. Tem-se w_i como o peso atribuído à fila correspondente. O peso relativo r_i da fila i é então definido como:

$$r_i = \frac{w_i}{\sum_{i=1}^I \rho_i} \quad (3.4)$$

Onde w_i é o peso atribuído da fila i e I é o número da classe.

O peso relativo pode ser compreendido como a probabilidade, em longo prazo, de uma classe ser servida quando todas as classes são continuamente colocadas em espera.

O WFQ usa apenas alguns campos do cabeçalho dos pacotes para determinar o protocolo subjacente. Para o protocolo IP, usam-se o *Type of Service* (ToS) e os endereços de origem e de destino. Para os protocolos TCP e UDP, usam-se as portas de origem e destino. (Kleber, 2009)

Como demonstrado na figura o escalonador é configurado em roteadores de bordas como foi especificado no estudo, assim estes vão se classificando e colocando em uma fila os pacotes que chegam.

4.2 EXEMPLO DO FUNCIONAMENTO WFQ

Para este, teremos como base a tese a tese do Prof. Kleber Nabas (2009).

No exemplo proposto por (MARQUES, 2005), o escalonamento WFQ é aplicado a três filas com cotas iguais de 8000 bits para cada classe de fluxo (as cotas são proporcionais aos pesos). No primeiro ciclo conforme é mostrado na Figura 20, todas as variáveis saldo são inicializadas com zero. Na lista de filas ativas um ponteiro indica para qual fila os recursos do sistema estarão disponíveis naquele momento. Neste caso os recursos do sistema estão disponíveis para a fila 1. Como o saldo de 8000 bits é insuficiente para transmitir o pacote de 12000 bits o saldo é acumulado para o próximo ciclo em que os recursos estarão disponíveis para esta fila e encerra-se o ciclo atual.

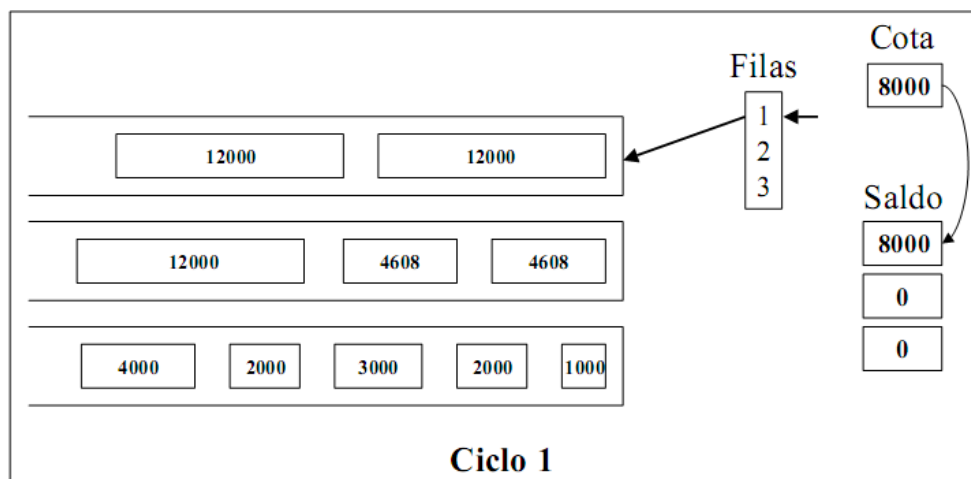


Figura 20: Ciclo 1 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 13

No ciclo 2 a fila ativa passa a ser a fila 2. Nesta fila com o saldo de 8000 bits é transmitido um pacote de 4608 bits. Desta forma, o saldo passa a ser 3392 bits e é acumulado para a transmissão do próximo pacote.

Veremos este ciclo na figura a seguir:

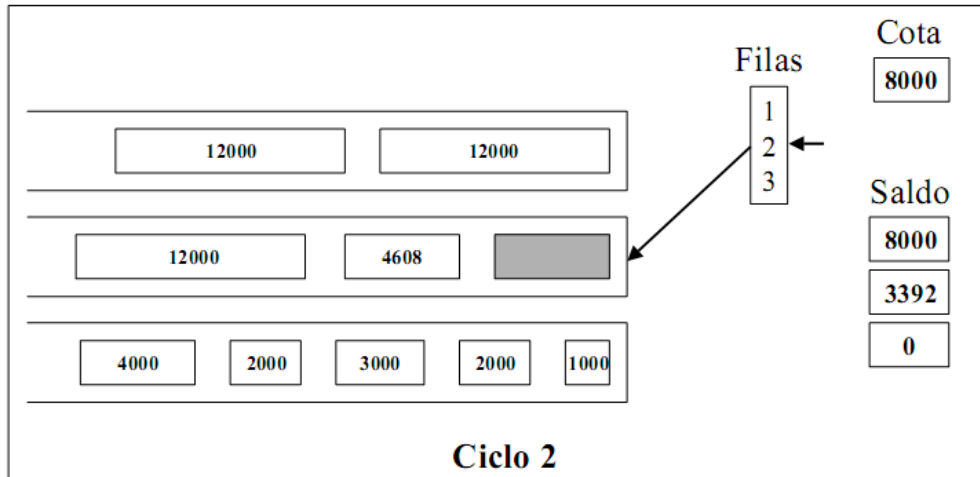


Figura 21: Ciclo 2 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 14

Já no ciclo 3 com a fila 3 ativa são transmitidos os pacotes de 1000 bits, 2000 bits, 3000 bits e 2000 bits que no total perfaz uma soma de 8000 bits. Dessa forma, o saldo é zerado e torna-se insuficiente para transmissão do último pacote presente nesta fila tendo como consequência o fim do ciclo corrente.

Este será ilustrado pela figura 22:

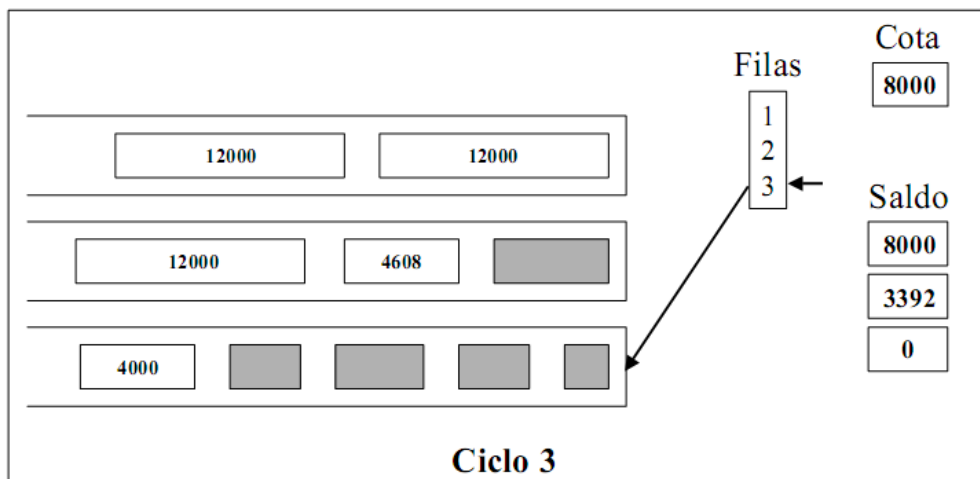


Figura 22: Ciclo 3 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 14

No ciclo 4 a fila ativa volta a ser fila 1. O saldo remanescente de 8000 bits é somado a cota de 8000 bits. Dessa maneira, o saldo passa a ser 16000 bits viabilizando a transmissão do primeiro pacote de 12000 bits. Após esta transmissão

o saldo passa a ser de 4000 bits e como não é possível a transmissão do último pacote de 12000 bits o saldo fica acumulado para o próximo ciclo desta fila.

A Figura 23 mostra a situação acima citada:

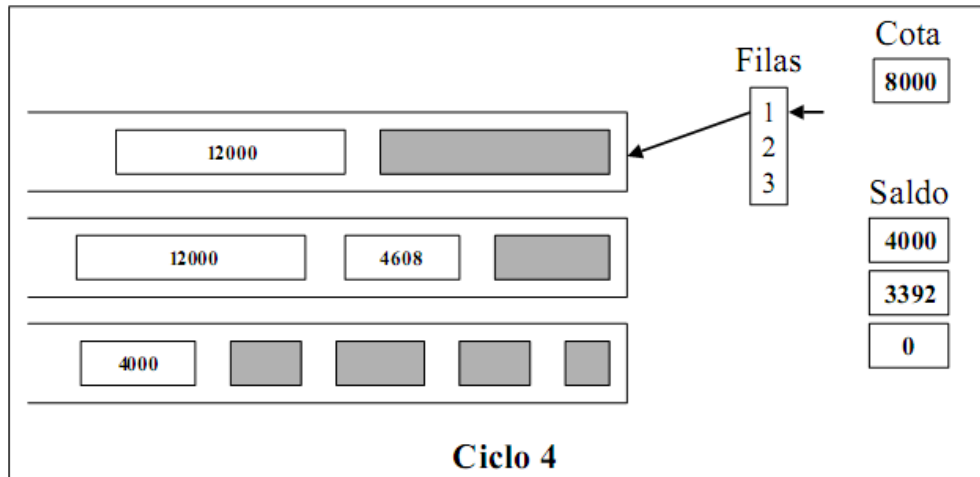


Figura 23: Ciclo 4 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 15

Na seqüência a fila ativa passa ser a fila 2. O saldo anterior é somado a cota de 8000 bits resultando num saldo de 11392 bits. Por meio deste saldo é transmitido mais um pacote desta fila. O saldo restante é 6784 bits e fica acumulado para o próximo ciclo desta fila uma vez que resta apenas mais um pacote na fila.

Acompanhemos na próxima figura o ciclo 5.

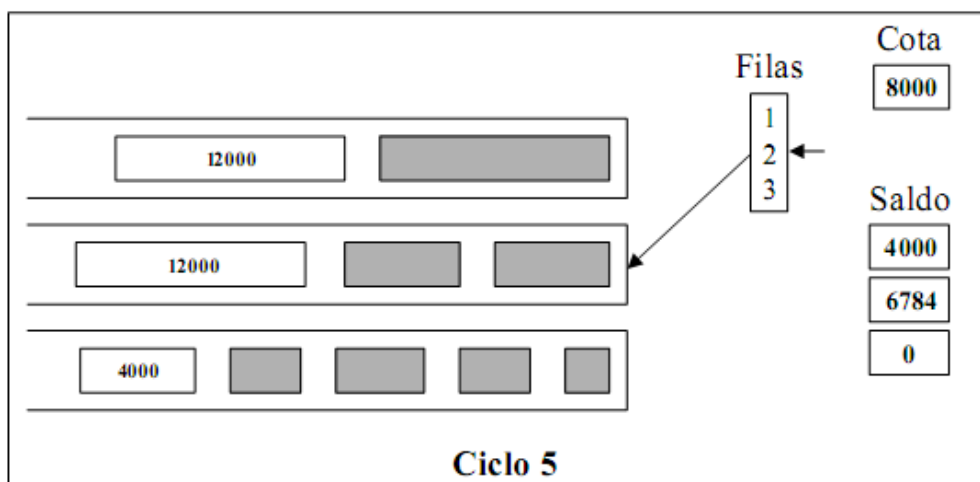


Figura 24: Ciclo 5 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 15

No ciclo 6 o saldo somado a cota denota um total de 8000 bits. Dessa forma, o pacote de 4000 bits é transmitido e como não há mais pacotes nesta fila o saldo é zerado e a lista de filas ativas passa a contar somente com as filas 1 e 2.

A próxima figura mostra o ciclo:

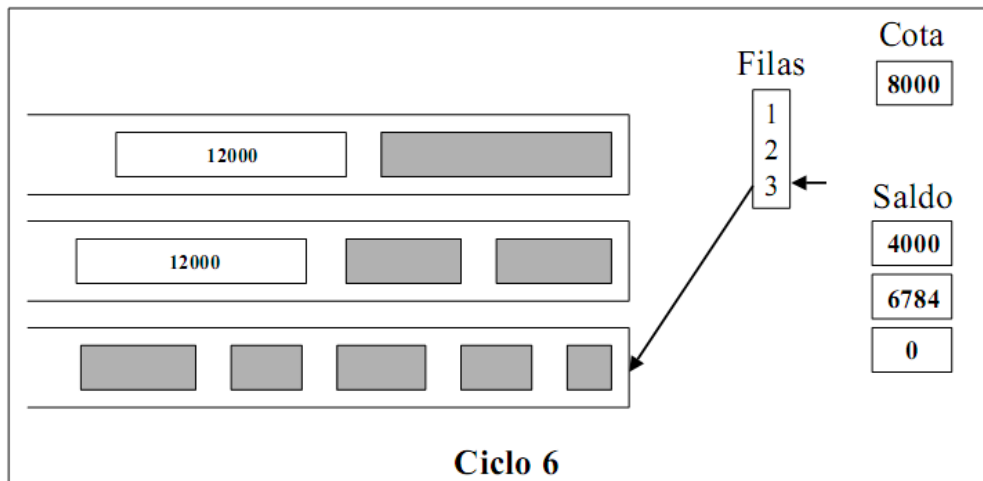


Figura 25: Ciclo 6 – WFQ
 Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 16

No ciclo 7 é transmitido o último pacote da fila 1 tendo em vista que o saldo de 12000 bits garante a transmissão do pacote. Sendo este pacote o último a ser transmitido o saldo é zerado e partir deste ciclo somente há uma fila com pacotes a serem transmitidos.

Veja o ciclo na figura 26:

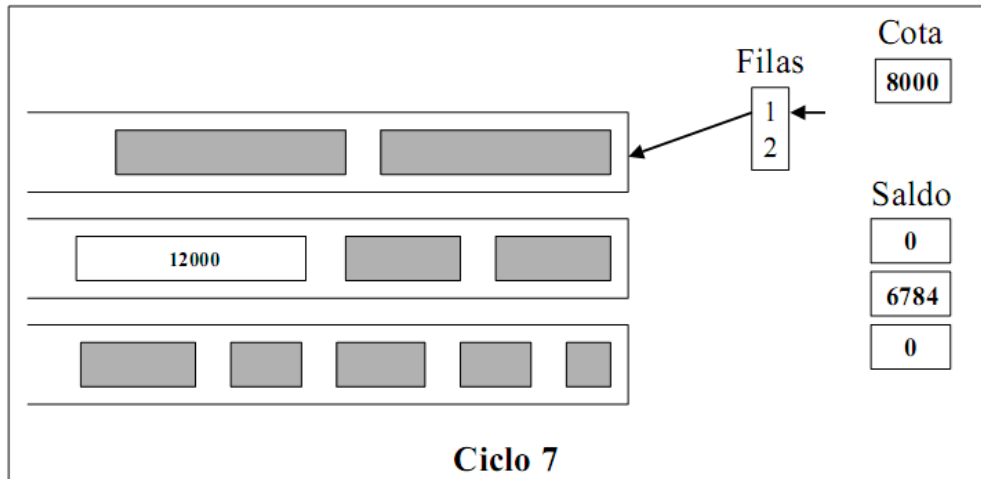


Figura 26: Ciclo 7 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 17

Por fim, o último pacote do sistema presente na fila 2 é transmitido. Desta forma, o saldo é zerado e nenhuma fila permanece ativa, pois, não há nenhum pacote no sistema. O sistema fica inoperante até o momento em que chegar algum pacote nas filas de espera podendo neste caso no máximo ter 3 filas ativas ao mesmo tempo.

Vamos acompanhar o modelo na figura seguinte:

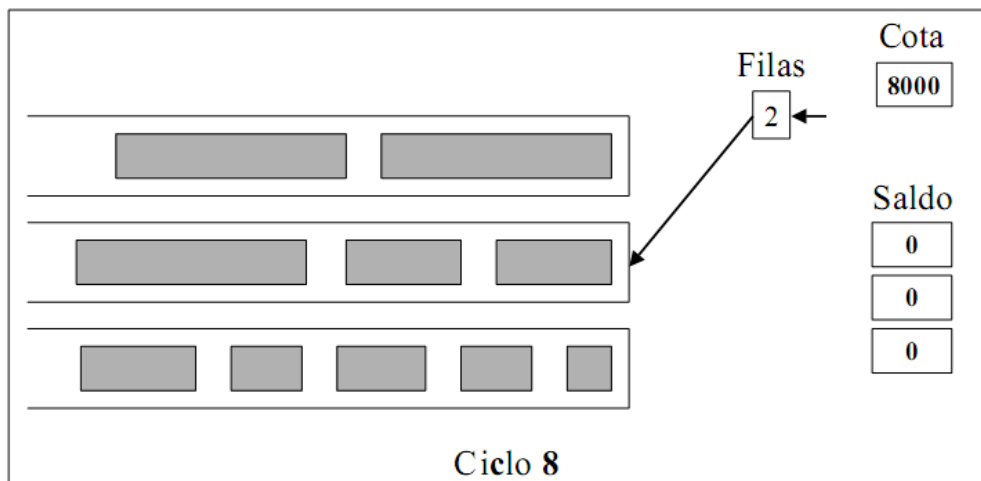


Figura 27: Ciclo 8 – WFQ

Fonte: Tese Prof. Kleber Nabas – 2009 – Capítulo 3 – Página: 17

5. CONCLUSÃO

Pode-se concluir através das pesquisas feitas para realizar este trabalho a importância da qualidade de serviço no tráfego em uma rede de dados.

A qualidade de serviço é definida através da escolha de um escalonador de pacotes para um nó de rede, este permite diferenciar o atraso entre os fluxos na rede de dados, fornecendo maior ou menor largura na banda e também podendo limitar o fluxo.

Nesse trabalho escolhemos o escalonador WFQ para demonstrar a qualidade que o mesmo oferece a rede de dados. Para definir esta, uso ainda a conclusão do Prof. Kleber Nabas:

“O escalonador utilizado foi o WFQ devido ao fato que o mesmo consegue atender a várias filas simultaneamente e diferenciá-las através do peso atribuído a cada uma delas. Foi mostrado um modelamento matemático do escalonador e um exemplo passo a passo do seu funcionamento.” [10].

O que facilita na escolha do escalonador de pacotes é que neste pode-se fazer a diferenciação dos fluxos mantendo a Qualidade de Serviço no tráfego de uma rede de dados.

6. REFERENCIAL BIBLIOGRÁFICO

[1] NABAS, Kleber K. H. **Arquitetura Para Provisão de QoS em uma Rede MPLS**. <http://inf.unisul.br/~ines/workcomp/cd/pdfs/2226.pdf>, 2009.

[2] PINHEIRO, José Maurício Santos. **Afinal o que é Qualidade de Serviço?** - http://www.projetoderedes.com.br/artigos/artigo_qualidade_servico.php. 2004. Arquivo consultado em julho de 2011.

[3] SOUZA, Sergio. **Aplicando políticas de QoS** - http://mum.mikrotik.com/presentations/BR08/Aplicando_politicas_de_QoS.pdf. Outubro/2008. Arquivo consultado em julho de 2011.

[4] AZEVEDO, Samuel Montalvão Varjão e ALMEIDA, Saulo Andrade – **QoS – Qualidade de Serviço** - www.logicengenharia.com.br/mcamara/ALUNOS/QoS.pps. Arquivo consultado em julho de 2011.

[5] FALSARELLA, Douglas. **Conceitos básicos de QoS** – (junho/2009) - <http://imasters.com.br/artigo/13011/redes-e-servidores/conceitos-basicos-de-qos>. Arquivo consultado em junho de 2011.

[6] CIESLAK – **Laboratório do Professor**. http://ccieslak.com/0-faculdade/1-unip/0-COMP-SIST-OPERACIONAL-2011/SO%20-%202011/so_4.pdf. Arquivo consultado em junho de 2011.

[7] SANTOS, Aline Guarato dos e PÓVOA, Marcelo H. M., - **Escalonamento de Processos – Universidade Federal de Uberlândia**. (2º semestre - 2005)

http://www.facom.ufu.br/~faina/BCC_Crs/INF10-2S2005/Download/ch5-p1-aline-marcelo.pdf. Arquivo consultado em junho de 2011.

[8] LEEK&LEIC – **Escalonamento do Processador** – Faculdade de Engenharia da Universidade do Porto - <http://pt.scribd.com/doc/7202387/Escalonamento-de-Processos>. Arquivo consultado em junho de 2011.

[9] TANENBAUM, Andrew S. – **Sistemas Operacionais Modernos** – São Paulo – 5ª reimpressão: Prentice Hall, 2003.

[10] NABAS, Kleber Kendy Horikawa. **Proposta de um modelo para análise de desempenho do Escalonador WFQ alimentado com tráfego LRD**. Tese CPGEI, UTFPR – 2009.

[11] PINA, Bernardo. CUNHA, Filipe. FERNANDES, Pedro. **Mecanismos de Escalonamento**. Faculdade de Ciências – Universidade do Porto. Departamento de Ciências de Computadores - <http://www.slideshare.net/pedronero/quality-of-service-and-linux>. Arquivo consultado em junho de 2011.

[12] JR, Peter Jandl. **Notas Sobre Sistemas Operacionais**. Apostila - <http://pt.scribd.com/doc/7051368/58/Escalonamento-FIFO-First-In-First-Out>. Arquivo consultado em Julho de 2011.

[13] KULCZYK, M. **Desenvolvimento de um simulador para escalonar o tráfego On/Off HTTP e FTP**. Monografia. (Bacharel em Ciência da Computação) – Universidade UNOCHAPECÓ – Chapecó, 2006.

[14] SILVA, Renato Moraes. **Avaliação de desempenho de fontes On/Off através do desenvolvimento de um simulador para escalonamento FIFO e DRR**. Monografia. (Licenciatura em Informática) Universidade Federal do Mato Grosso – 2009.