

Protótipo de um Simulador Multimídia para o Ensino de Escalonamento FIFO

Rodrigo Aparecido Morbach¹, Tatiana Annoni Pazeto²

¹UFMT – Universidade Federal do Mato Grosso
Licenciatura Plena em Informática.
Rodovia Rondonópolis – Guiratinga, KM 06
(MT-270) Bairro Sagrada Família
CEP 78735-910, Rondonópolis, MT.

¹rodrigo_mtroo@hotmail.com
²tpazeto@gmail.com

Resumo. *Este artigo tem por objetivo apresentar uma ferramenta desenvolvida para o ensino de escalonamento de pacotes em redes de computadores. Através da pesquisa realizada, verificou-se que dentre os softwares utilizados no ensino de redes, não se encontrou nenhum específico para escalonamento, tão pouco utilizando a abordagem VAK (Visual, Auditory and Kinesthetic), proposta para esta ferramenta. Assim, a partir da análise do simulador desenvolvido por Renato Silva em 2009, foi implementado o REDESIM, que apresenta uma interface baseada no estilo VAK, gerando resultados das simulações de forma animada e que permite análises mais rebuscadas, usando editores de planilhas. Também possibilita a atualização automática dos dados gerados, conforme a realização de novas simulações. Deste modo, constatou-se que o REDESIM atende a abordagem VAK, pois com o uso dos recursos multimídias é possível atender a usuários que assimilam mais facilmente o conhecimento de modo visual, auditivo ou sinestésico.*

Palavras-chave: *Redes, Simulação, Ferramentas educacionais*

1. Introdução

O escalonamento de pacotes é um processo realizado por um algoritmo implementado em dispositivos de rede. Seu intuito é criar uma fila para o atendimento de pacotes, de tal modo que evite congestionamentos e conseqüentemente atrasos, além da perda de pacotes. Esta última pode acarretar no reenvio de pacotes, ocasionando prejuízos ao sistema

Para atender as diversas necessidades de usuários e aplicações em um ambiente de redes, existem vários algoritmos como o *First-In First-Out (FIFO)*, *Deficit Round Robin (DRR)*, *Weighted Fair Queuing (WFQ)*, *Class Based Queuing (CBQ)*, entre outros. Estes possuem características diferentes que os tornam ideais para certas condições de tráfego, sendo que há alguns com funcionamento simples, como o FIFO. Outros são mais complexos, permitindo atender os pacotes utilizando outros critérios além da ordem de chegada dos pacotes. Assim, os mesmos podem garantir a Qualidade de Serviço (QoS) requisitada pelas aplicações.

Neste sentido, os conceitos envolvidos no processo de escalonamento podem ser difíceis de visualizar, principalmente se o assunto for estudado apenas através de textos ou aulas expositivas e dialogadas.

Visando preencher esta lacuna, esta pesquisa tem por objetivo desenvolver uma ferramenta de auxílio ao ensino de escalonamento de pacotes, utilizando os estilos de aprendizagem *Visual, Auditory and Kinesthetic* (VAK). Desta forma, a ferramenta proposta deverá explorar o uso de recursos multimídia como, animação, sons, imagens e vídeos, de modo a atender aos diferentes estilos de aprendizagem.

Para o desenvolvimento da ferramenta proposta, foi utilizado como base o simulador de escalonamento FIFO desenvolvido por Renato Silva [1]. A este simulador foram adicionadas características para sanar as finalidades educacionais. Assim, este trabalho apresenta as características do escalonador FIFO, assim como as fontes geradoras de tráfego On/Off Exponencial, uma vez que estes são conceitos utilizados no trabalho de Renato Silva [1], o qual esta pesquisa dá prosseguimento.

O trabalho está organizado em seis seções. Na seção dois é realizada uma introdução aos algoritmos de escalonamento, sendo detalhado o escalonamento FIFO, bem como é apresentado o simulador criado por Renato Silva [1], o qual serviu de base para a implementação do simulador proposto neste trabalho. Na seção três é enfatizada a fonte geradora de tráfego On/Off Exponencial, modelo usado no desenvolvimento das fontes do trabalho de Renato Silva [1]. Na seção quatro, são apresentados aspectos educacionais que embasaram o desenvolvimento da pesquisa. Na seção cinco é mostrado o REDESIM, onde também são relatados os resultados gerados. Na seção seis estão as considerações finais, seguido das referências.

2. Algoritmos de Escalonamento

Segundo Zweben e Fox [2], escalonamento é:

[...]” A seleção entre planos alternativos alocando recursos e atividades em cada instante de tempo, tal que esta designação obedeça as restrições temporais das atividades e as limitações de capacidade de um conjunto de recursos compartilhados”.

Segundo Comer [3, p.102] “um dos fatos que motiva o uso de pacotes é que, por exemplo, um remetente e um receptor precisam coordenar a transmissão para assegurar que os dados cheguem corretamente ao seu destino, pois, quando ocorrem erros de transmissão alguns dados podem ser perdidos. A divisão dos dados em pequenos blocos ajuda o remetente e o receptor a determinar quais blocos chegaram intactos e quais não chegaram”. O uso de pacotes pequenos e algoritmos responsáveis por fazer o escalonamento destes, ajuda a assegurar a justiça no acesso.

Desta forma, como afirma Chun e Baker [4] “[...]a escolha de um algoritmo de escalonamento para determinar quais serão os próximos pacotes a serem processados pode ter um efeito significativo, dependendo do tipo de tarefa realizada”.

2.1. First-In First-Out (FIFO)

Este algoritmo trata os pacotes de acordo com a ordem de chegada, sendo um dos mais simples quanto à implementação

Segundo Wrege *et al* [5], um escalonador de pacotes para atender uma qualidade de serviço (QoS) de rede deve ser sofisticado o suficiente para suportar rigorosas restrições de QoS com cargas elevadas para que os pacotes possam ser tratados sem perdas significativas. Desta forma, o FIFO não é recomendado para aplicações que exijam QoS, pois ele não permite priorizações. Contudo, sua simplicidade computacional faz com que ele seja um dos mais implementados nos equipamentos de interconexão [6]. Além disso, uso do escalonamento FIFO nesse trabalho se deve ao fato de ele ter sido desenvolvido no trabalho de Renato Silva [1].

2.2 O Simulador Desenvolvido por Renato Silva [1]

A primeira etapa para o desenvolvimento do REDESIM foi o levantamento de requisitos. Este foi realizado baseando-se na interface e funcionalidades do simulador desenvolvido por Renato Silva [1], cuja tela completa é apresentada na Figura 1.

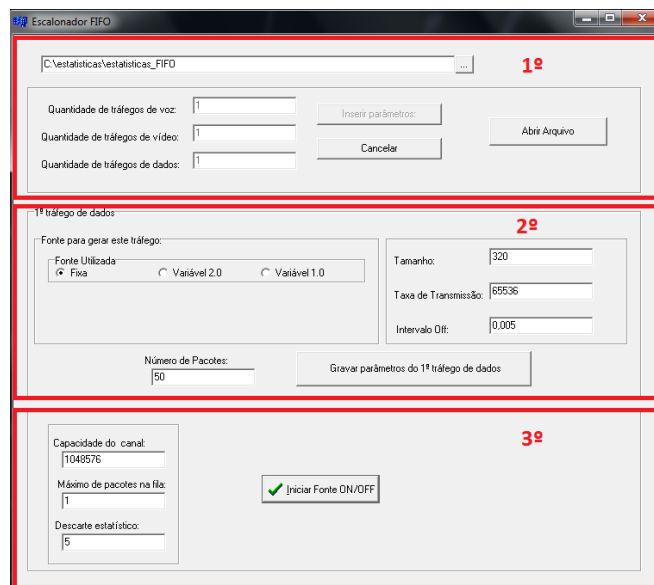


Figura 1. Simulador de escalonamento de tráfego FIFO de Renato Silva [1].

Como pode ser visto na Figura 1, o simulador desenvolvido possui uma série de parâmetros que precisam ser conhecidos antes de sua utilização, o que torna seu uso bastante restrito. À medida que estes parâmetros vão sendo preenchidos novos campos de entrada vão surgindo, conforme a sequência mostrada na Figura 1.

O simulador desenvolvido por Renato Silva [1] tinha por objetivo fazer a avaliação de desempenho das fontes On/Off propostas, ou seja, qual seria o seu impacto no escalonador FIFO. Assim, tendo como base o software proposto, teve-se a ideia de adaptar o simulador para que este pudesse ser utilizado para fins educacionais, acrescentando características que contribuíssem nesse processo.

3. Fontes de tráfego On/Off Exponencial

Neste modelo de geração de tráfego há dois intervalos: On, em que são gerados pacotes, e o intervalo de Off, período em que não há geração de tráfego, mas que pode ser usado para transmissão de informações de sinalização e controle. Este modelo pode ser representado através de uma cadeia de Markov de dois estados, On e Off, e ainda duas taxas de transição, α , que corresponde a taxa de transição de On para Off, β , que

corresponde a taxa de Off para On, [7]. O modelo On/Off exponencial é muito popular por ser simples e genérico para gerar tanto voz, quanto vídeo e dados.

A seguir serão apresentadas as fontes desenvolvidas por Renato Silva [1], baseadas no modelo On/Off Exponencial.

3.1. As fontes utilizadas no trabalho

No trabalho de Renato Silva [1] foram desenvolvidas três fontes de tráfego denominadas: Fixa, Variável 1.0 e Variável 2.0. A fonte Fixa possui como parâmetros de entrada o tamanho do pacote, que é fixo, a taxa de transmissão, o intervalo de Off e o número de pacotes que serão gerados. Além disso, nesta fonte é gerado um pacote a cada intervalo de On, intervalo esse obtido dividindo o tamanho do pacote pela taxa de pico da fonte em questão.

A fonte On/Off variável 1.0 possui alguns parâmetros de entrada em comum com a fonte Fixa. No total, os parâmetros usados são: o número de pacotes a serem gerados; tamanho do pacote, que neste caso tem seu tamanho gerado com distribuição exponencial; taxa de transmissão; intervalo de Off; quantidade de pacotes que devem ser gerados por intervalo; e a opção de gerar intervalos com quantidade de pacotes fixa ou média.

A fonte On/Off variável 2.0 possui todos os parâmetros de entrada da fonte On/Off variável 1.0. Contudo, ela difere do modelo anterior que gerava uma quantidade de pacotes e a partir disso calculava o tempo de On necessário para transmitir esses pacotes gerados. Assim, nesta fonte é gerado o intervalo de On seguido da geração de um pacote. Então é verificada a possibilidade de enviá-lo no intervalo ativo corrente. Caso seja possível, o pacote é enviado. Caso contrário, o pacote é armazenado para ser enviado no próximo intervalo.

4. Fundamentos para o desenvolvimento da aplicação

A seguir serão apresentadas as informações que motivaram e embasaram esta pesquisa.

4.1. Os Diferentes Estilos de Aprendizagem

Segundo Melis [8], para um ambiente de aprendizagem ser eficaz, deve capitalizar não só o contexto geral, mas as características de cada aluno.

É fato conhecido que muitos estudantes possuem preferências ao estudar e aprender um conteúdo. Muitos gostam de usar marcadores de texto para destacar pontos importantes, outros costumam gravar as aulas para ouvir em casa, há também aqueles que precisam por em prática a teoria para aprender algo, principalmente na área de ciências exatas. Por vários autores isto é chamado de estilo de aprendizagem, ou seja, o conjunto de preferências de uma determinada pessoa em utilizar formas diferenciadas de memorizar ou aprender algo.

O estilo de aprendizagem VAK se baseia no uso dos três canais sensoriais: visão, audição e sinestesia (tato e movimento). Willingham [9, p. 03] justifica o uso da aprendizagem VAK em sala de aula pelo fato de que os alunos aprendem mais quando o conteúdo é apresentado a sua maneira, “o que pode maximizar ainda mais o aprendizado com aulas diferentes para cada tipo de aluno. Por exemplo, dentro de uma classe do jardim de infância, o aluno auditivo poderia ouvir histórias sobre as férias em diferentes

locais do mundo. O aluno visual pode examinar fotos de celebrantes de férias, enquanto o aluno sinestésico verifica trajes e artefatos associados às férias”.

Considerando essas premissas, desenvolver um sistema multimídia que possa contemplar os três canais receptores, de acordo com o sistema VAK, pode melhorar e facilitar a aprendizagem do aluno, visto que em um ambiente desse tipo pode-se ter tanto imagens, sons, vídeos, animações, quanto uma interface que possibilite ao aluno interagir com o sistema de forma prática.

4.2. O Ensino da Disciplina de Redes

Para Chen [10, p. 01], “um entendimento de redes de computadores é cada vez mais importante em um mundo dos negócios de alta tecnologia. Desta forma, faculdades e universidades precisam trabalhar conceitos de redes de computadores para que os alunos sejam capazes de se comunicar de forma inteligente como profissionais”.

Chen [10] relata que quando iniciou o ensino da disciplina de redes no nível superior, sempre ministrava suas aulas de maneira convencional, com slides, material impresso, dentre outros. Logo percebeu que os alunos ainda encontravam muitas dificuldades para compreender o conteúdo, e, a partir daí, começou a fazer uso de trabalhos práticos, através dos quais se verificou o aumento no nível de compreensão da turma.

Diante disso, fica evidente a necessidade de construção de objetos de aprendizagem para a visualização e auxílio educacional. Contudo, para que a construção destes objetos tenha alguma fundamentação, é importante analisar o que existe a disposição no mercado. Isso aguça a criatividade, bem como auxilia na compreensão do que pode atender ao estilo de aprendizagem VAK.

4.3. Ferramentas para o ensino de redes

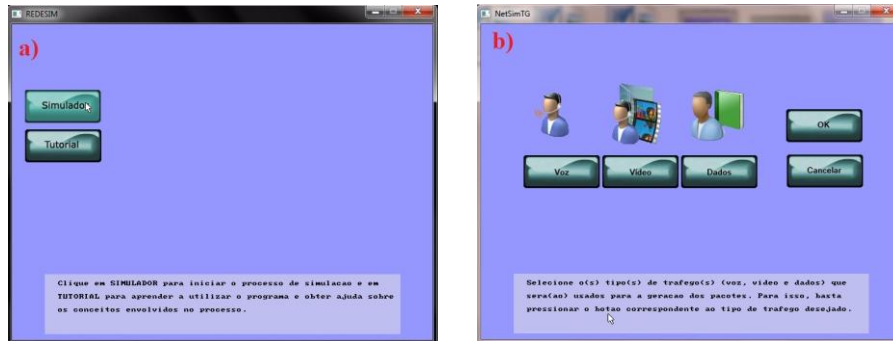
Existem vários softwares que podem ser utilizados para fins de estudo de redes de computadores. Sarkar [11] apresenta alguns dos mais conhecidos, a citar: *Network Simulator*, *WebLan-Designer*, *DlpSim*, *WebTrafMon*, *OMNet*, dentre outros. Realizada uma análise nos softwares supracitados e outros presentes no trabalho de Sarkar [11], pode-se concluir que, apenas o *Network Simulator* abrange o conteúdo de escalonamento de pacotes. Além disso, o mesmo requer um conhecimento considerável para a sua utilização.

Embora existam diversos recursos para a aprendizagem, poucos podem contribuir com alunos sinestésicos, por possuírem, em sua maioria, somente recursos visuais. Neste sentido, um tipo de software que atende este grupo são os simuladores.

5. REDESIM: O Simulador Proposto

A partir do levantamento e análise de requisitos realizados com base na interface e funcionalidades do simulador desenvolvido por Renato Silva [1], foi proposta uma interface com o objetivo de que o programa pudesse ser utilizado por estudantes de computação ou áreas afins, com pouco conhecimento no assunto. Assim, na Figura 2 é mostrada a tela inicial do programa REDESIM, onde o usuário pode acessar direto a seção de configuração de parâmetros para a simulação, através do botão “Simulador”, ou também poderá optar por aprender sobre os conceitos envolvidos na simulação,

como fontes de tráfego, escalonamento de pacotes e os parâmetros envolvidos no processo. O tutorial ainda não está em funcionamento nessa versão do simulador, porém, para trabalhos futuros, pretende-se que seja apresentado no formato HTML compilado (.CHM), onde serão inseridos textos, imagens e vídeos ou animações, de forma a atender aos diferentes estilos de aprendizagem e principalmente usuários auditivos.



(a) Tela inicial do programa (b) Tela de escolha dos tipos de tráfegos

Figura 2. Tela principal e Tela de escolha de tráfegos

Após clicar no botão “Simulador”, apresentado na Figura 2 “a”, o usuário será direcionado para a tela mostrada a direita, representada pela letra “b”, onde se deve escolher quais os tipos de pacotes trafegarão na rede durante a simulação, sendo as opções, voz, vídeo ou dados.

Depois de escolher os tipos de tráfego que serão utilizados na simulação e clicar no botão “Ok”, o programa exibirá uma caixa de texto onde o usuário deverá entrar com a quantidade de tráfego, sendo esta operação realizada para cada um dos tipos escolhidos (voz, vídeo ou dados). Este procedimento é apresentado na Figura 3, onde mostra a entrada para usuários de voz.



(a) Tela de inserção da quantidade de usuários de tráfegos (b) Tela de inserção dos parâmetros de geração dos pacotes

Figura 3. Quantidade de usuários

Após inserir a quantidade de usuários de voz, conforme mostrado na Figura 3, será exibida uma tela onde o usuário deverá digitar os parâmetros para cada um dos usuários de voz, vídeo ou dados. Por exemplo, se na Figura 3 “a” foi inserido dois usuários de voz, isso significa que na Figura 3 “b” o usuário do programa deverá entrar com os parâmetros mostrados duas vezes, sendo uma vez para cada usuário de voz. Além disso, o usuário poderá escolher entre as fontes Fixa, variável 1.0 e variável 2.0, sendo que,

conforme a fonte escolhida haverá parâmetros diferentes a serem preenchidos. É importante ressaltar que a velocidade de transmissão varia de acordo com os parâmetros configurados.

Em seguida, após escolher a fonte geradora de tráfego para cada usuário e preencher todos os parâmetros, o programa irá exibir a tela de animação da simulação, mostrada na Figura 4, onde há três computadores, cada um representando um tipo de tráfego (voz, vídeo e dados) gerando pacotes (representados por envelopes) até o servidor da rede. No canto superior direito da tela, são exibidos os tempos de chegada do último pacote de cada tipo de tráfego.

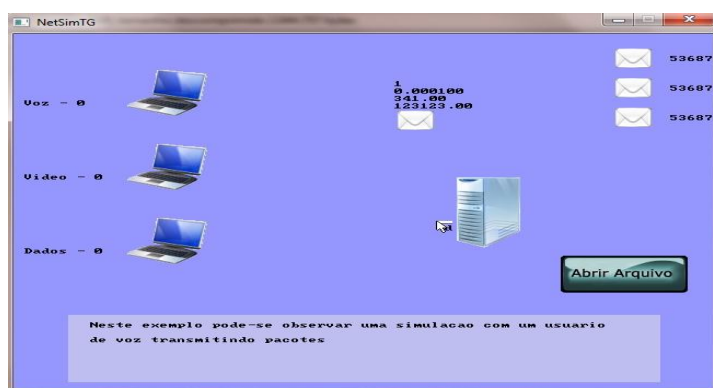


Figura 4. Tela que mostra a animação do processo de escalonamento

A animação da Figura 4 mostra os pacotes sendo transmitidos para o servidor na mesma ordem em que foram inseridos durante o programa. Nas informações numéricas podem ser verificados o índice do pacote, a taxa de transmissão, intervalo de off e o tamanho do pacote, todos esses definidos pelo usuário durante a inserção dos parâmetros.

O simulador proposto atende ao estilo VAK, pois os usuários visuais são agraciados através de animação, onde eles poderão visualizar como é feito o envio das informações preenchidas nas telas anteriores. Para usuários sinestésicos, o simulador oferece a possibilidade de fazer vários testes com informações diferenciadas, permitindo ao usuário fazer uma análise criteriosa sobre as fontes e os tipos de tráfego. Contudo, o simulador ainda não contempla nenhum recurso para os usuários auditivos. Isso será realizado através do tutorial que será disponibilizado na próxima versão.

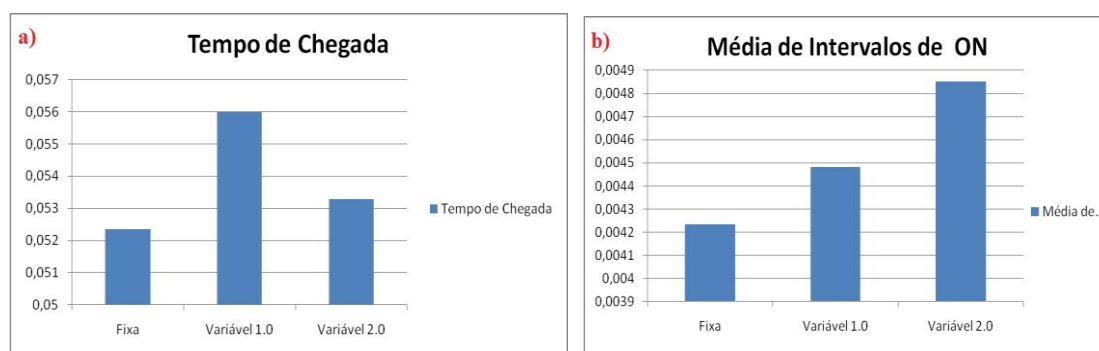
Terminada a animação, o usuário deve clicar em “Abrir Arquivo”. Através deste procedimento, o programa irá abrir o editor de planilhas instalado na máquina. Este apresentará os dados gerados pelo simulador em formato *txt*. Uma vez carregados os dados, o editor de planilhas gera automaticamente os gráficos comparativos entre as fontes geradoras de tráfego, permitindo desta forma a análise dos dados. Além disso, apresenta informações que possibilitam verificar o impacto causado pelos parâmetros digitados pelo usuário em relação à fonte geradora de tráfego escolhida e em função do servidor onde se encontra o escalonador. Estas informações são o tempo de fila, tempo de sistema, tempo de serviço e descarte de pacotes.

5.1. Análise dos dados gerados

A partir das informações do simulador são gerados gráficos para comparar o tempo de chegada do último pacote de cada fonte, a média do período On, dentre outras informações. Essa comparação é importante, pois permite visualizar que, dependendo do tipo de serviço requisitado, determinado formato de pacote será mais adequado.

Na geração do gráfico comparativo de tempo de chegada é selecionada a célula correspondente ao tempo de chegada do último pacote de cada fonte no servidor, onde o gráfico mostra o título da comparação, e as barras que correspondem ao tempo de chegada do último pacote de cada fonte. Este gráfico está apresentado na Figura 5 “a”.

Para a geração do gráfico correspondente as médias de On de cada fonte também foi necessário selecionar em cada estatística os valores correspondentes. Vale ressaltar que isto só é feito uma vez, pois os dados são gerados automaticamente quando o documento é aberto. Assim toda vez que os dados forem alterados os valores dos gráficos também sofrerão alteração de acordo com as fontes. O resultado final dos gráficos representando o tempo de chegada e as médias de intervalos de On é apresentado na Figura 5 “b”.



(a) Comparação de tempo de chegada

(b) Média de Intervalor de On

Figura 5. Telas de Geração de gráficos

Através da Figura 5 “a” percebe-se que o tempo de chegada do último pacote da Variável 1.0 ao servidor foi o maior dentre os três. Isso se deve tanto ao tamanho do pacote definido no momento da inserção dos parâmetros, quanto a taxa de transmissão definida para esta fonte.

O gráfico comparativo das médias de On de cada fonte mede o tempo em que o programa esteve no estado ativo, ou seja, gerando pacotes. Esse tipo de comparação permite visualizar qual das fontes permaneceu mais tempo em funcionamento, e qual deixa o servidor mais tempo ocioso. Na Figura 5 verifica-se que a fonte Variável 2.0 teve sua média de tempo de On mais elevada. Esta foi a que permaneceu mais tempo gerando pacotes, ficando menos tempo em estado de silêncio.

Na mesma planilha de texto também é realizada uma comparação entre as médias do intervalo de Off. Como os gráficos são apresentados na mesma planilha, o usuário pode fazer a comparação das três fontes em diferentes perspectivas. Futuramente pretende-se incluir alguns exercícios com opções de multipla escolha para que o usuário possa testar o conhecimento adquirido, e, ao fim do teste seria apresentado um contador com o número de acertos.

Além da comparação entre as fontes, a aplicação proposta faz uma análise das mesmas usando o escalonador FIFO. Assim, o usuário deve definir também a capacidade do canal e o tamanho do buffer.

Neste aspecto a aplicação permite analisar o tempo de sistema, que corresponde ao somatório do tempo de serviço e de fila, bem como o descarte de pacotes. Vale ressaltar que o tempo de serviço é equivalente ao tempo de processamento do pacote, correspondendo ao tamanho do pacote dividido pela taxa de transmissão, e o tempo de fila que é o tempo que o pacote esperou para ser atendido. Um exemplo de gráfico gerado para o escalonamento FIFO é apresentado na Figura 6.

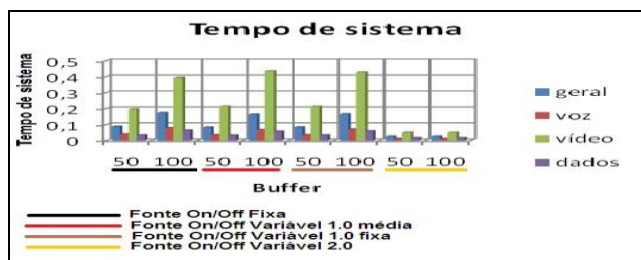


Figura 6. Comparação entre o tempo de sistema das diferentes fontes

Pode-se observar que o tempo de sistema é bem superior com o buffer de 100 posições do que com 50 posições. Isso ocorre porque quanto maior o tamanho do buffer, menos pacotes são descartados. Logo o número de pacotes processados é maior.

Com relação ao tipo de tráfego, também nota-se uma grande diferença no tempo de sistema. Isso acontece porque os pacotes de dados chegam com maior frequência ao servidor, acarretando em um tempo de fila superior aos tráfegos de voz e vídeo. Isso em função dos parâmetros usados para a geração dos tráfegos de voz, vídeo e dados. Assim, observa-se que o tamanho do pacote influencia diretamente no desempenho do escalonador.

6. Considerações Finais

Através das pesquisas realizadas foi possível desenvolver uma ferramenta de auxílio ao ensino de escalonamento, o REDESIM, baseado no simulador de escalonamento FIFO desenvolvido por Renato Silva [1].

Desta forma, o REDESIM atende aos princípios de aprendizagem VAK, visto que o usuário pode identificar os tipos de dados tratados através das imagens inseridas no programa e da animação. Atende também o princípio sinestésico, uma vez que todos os parâmetros são definidos pelo usuário, gerando impacto nos resultados, possibilitando comparações e visualização da fonte que mais se adequa a determinada situação. Futuramente pretende-se acrescentar ao simulador REDESIM efeitos sonoros para que se possa atender de maneira completa ao princípio de aprendizagem VAK, aprimorando ainda mais o ensino de escalonamento de redes.

Um dos principais resultados foi a criação de formas de análise dos resultados gerados, sendo esta baseada em gráficos, os quais permitem visualizar o impacto causado pelos parâmetros digitados pelo usuário.

Para trabalhos futuros pretende-se implementar o mesmo método utilizado para a criação do REDESIM para a criação de um simulador de escalonamento DRR baseado

em softwares livres, e ainda, procurar novas formas para a avaliação do aprendizado, sendo uma delas através da aplicação de exercícios.

7. Referências

- [1] Silva, Renato Moraes. Avaliação de Desempenho de Fontes On/Off Através do Desenvolvimento de um Simulador para Escalonamento FIFO E DRR. Monografia (Licenciatura Plena em Informática). Universidade Federal de Mato Grosso, Rondonópolis, 2009.
- [2] Zweben, M; Fox, M.S. Intelligent Scheduling. São Francisco-Califórnia, Morgan, Kaufmann, 1994
- [3] Comer, Douglas E. Computer Networks and Internets with Internet Applications, 4^a Department of Computer Sciences Purdue University. Editora Bookman, 2009.
- [4] Chun, B.; Baker, Mary. Evaluation of Packet Scheduling Algorithms in Mobile Ad Hoc Networks. Computer Science Department, Stanford University, Califórnia, 2002.
- [5] Wrege, Dallas E; Liebeherr, Jorg. A Near-Optimal Packet Scheduler for QoS Networks. Proceedings of the IEEE. Department of Computer Science, University of Virginia, Charlottesville, 1997.
- [6] Duan, Zhenhai; Zhang, Zhi-Li; Hou, Yimei Thomas. Fundamental Trade-Offs in Aggregate Packet Scheduling. IEEE Transaction On Parallel and Distributed Systems, Vol. 16, NO. 12, 2005.
- [7] Veríssimo, Fernando. C. A (2005). Propostas e Avaliações de Protocolos de Acesso Alternativos ao Padrão IEEE 802.11e. 2005. 124 f. Tese (Doutorado em Engenharia de Sistemas e Computação) – Tese, Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2005.
- [8] Melis, Erica. They Call It Learning Style But It's So Much More. German Research Institute for Artificial Intelligence. University of London. Saarbrücken, German, 2006. Disponível em:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.4062&rep=rep1&type=pdf>. Acesso dia 22-01-2011.
- [9] Willingham, Daniel T. Do Visual, Auditory, and Kinesthetic Learners Need Visual, Auditory, and Kinesthetic Instruction?. Reading rockets: Teaching for kids to read help those who struggle, 2006. Disponível em:
<http://www.readingrockets.org/article/12446>. Acesso dia 23-03-11.
- [10] Chen, Catherine. A Constructivist Approach to Teaching: Implications in Teaching Computer Networking. Information Technology, Learning, and Performance Journal, Vol. 21, No. 2, Fall 2003.
- [11] Sarkar, Nurul I. Tools for Teaching Computer Networking and Hardware Concepts. Hershey, PA, USA. Information Science Publishing. 2006.