

7

Referências Bibliográficas

[1] Internet World Stats, “Internet World Usage”, <http://www.internetworldstats.com/stats.htm>, Acessado em 14-05-2009.

[2] L. Badia, M. Lindstrom, J. Zander, M. Zorzi, “Demand and Pricing Effects on the Radio Resource Allocation of Multimedia Communication Systems”, IEEE Global Telecommunications Conference, GLOBECOM'03, 2003.

[3] L. Badia, M. Zorzi, “On Utility based Radio Resource Management with and without Service Guarantees”, in Proceedings of ACM MSWiM'04, 2004.

[4] L. Badia, M. Lindstrom, J. Zander, M. Zorzi, “An Economic Model for the Radio Resource Management in Multimedia Wireless Systems”, Proceedings of ASWN 2003, pp 121–133, 2003.

[5] R. Ribeiro, R. M. Salles, “Mecanismos para Controle de Tráfego UDP através de Política de Preços Baseada na Utilidade”, XXV Simpósio Brasileiro de Telecomunicações, 2007.

[6] M. O. Monteiro, M. A. Grivet, “Tarifação Dinâmica em Redes celulares”, XXI Simpósio Brasileiro de Telecomunicações, 2004.

[7] N. C. Ericsson, “Revenue maximization in resource allocation: Applications in wireless communication networks”, PhD Thesis, Uppsala University, Uppsala, Sweden, Oct. 2004.

[8] R. A. Oliveira Neto, F. P. Cavalcanti, “Aspectos Econômicos no Gerenciamento de recursos da rede radio”, XXV Simpósio Brasileiro de Telecomunicações, 2007.

[9] A. B. Mackenzie, L. DaSilva, Game Theory for wireless engineers, Morgan & Claypool publishers, 1 ed, 2006.

[10] Hal. R. Varian, Princípios Básicos de Microeconomia, Campus, 6 ed., 2003.

- [11] Kun I. Park, QoS in Packet Networks, Springer, 2005.
- [12] Jun Xu, Richard J. Lipton, “On Fundamental Tradeoffs between Delay Bounds and Computational Complexity in Packet Scheduling Algorithms”, in Proceedings of ACM SIGCOMM '02, August 2002
- [13] D. Stiliadis, A. Varma, “Rate-proportional Servers: A Design Methodology for Fair Queueing Algorithms”, IEEE/ACM Transactions on Networking, vol. 6, No. 2, pp. 164–174, 1998.
- [14] Internet Engineering Consortium, “Voice over Internet Protocol”, <http://www.iec.org/online/tutorials/vfoip/topic02.asp>, Acessado em 10-05-2009.
- [15] VoIP Trouble Shooter, “Jitter Buffer Configuration”, www.voiptroubleshooter.com/problems/jbconfig.html, Acessado em 15-05-2009.
- [16] M. J. Morrow, A. Sayeed, “MPLS and Next Generation Networks: Foundations for NGN and Enterprise Virtualization”, Cisco Press, 1 ed, 2006.
- [17] A. H. Schaaf, “NGN Challenges”, in Proceedings of World Telecommunications Congress, May 2006.
- [18] S. E. Kingsland, “Modeling Nature”, University of Chicago Press, 2. Ed, 1995.
- [19] A. F. V. Macedo, “Disciplinas de Serviço em redes de comunicações”, Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2001.
- [20] C. Semeria, “Supporting Differentiated Service Classes: Queueing scheduling disciplines”, Technical Paper, Juniper, 2001.
- [21] M. Gospodinov, “The affects of different queueing disciplines over FTP, Video and VoIP performance”, in Proceedings of CompSysTech, 2004.
- [22] Hui Zhang, “Service disciplines for guaranteed performance service in packet switching networks”, Proceedings of IEEE Vol.83, No. 10, October 1995.
- [23] A. Demers, S. Keshav, S. Shenker, “Analysis and simulation of a fair queueing algorithm”, in Proceedings of ACM SIGCOMM'89, pp 3-12.
- [24] S. J. Golestani, “A self-clocked fair queueing scheme for high speed applications,” in Proc. INFOCOM'94, Apr. 1994, pp. 636–646.
- [25] L. Zhang, “VirtualClock: A new control algorithm for packet switching networks,” in Proc. ACM SIGCOMM'90, Aug. 1990, pp. 19–29.

- [26] J. C. R. Bennett, H. Zhang, “WF2Q: Worst-case fair weighted fair queuing,” in Proc. INFOCOM’96, Mar. 1996, pp. 120–127.
- [27] P. Goyal, H. M. Vin, and H. Cheng, “Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks”, in Proc. ACM SIGCOMM’96, Aug. 1996, pp. 157–168.
- [28] M. Shreedhar, G. Varghese, “Efficient fair queuing using deficit round robin”, in Proc. ACM SIGCOMM’95, 1995, pp. 231–242.
- [29] S. Ramabhadran, J. Pasquale, “The Stratified Round Robin Scheduler: Design, Analysis and Implementation”, in IEEE/ACM Transactions on Networking, vol. 14, no.6, December 2006.
- [30] A. Adas, “Traffic Models for Broadband networks”, IEEE Communications Magazine, vol. 35, July 1997.
- [31] Recomendação 3GPP NOKIA “3GPP TSG-RAN WG4 (Radio) Meeting #39, R4-060438”, Shanghai, China, May 8-12, 2006.
- [32] A. K. Parekh, R. G. Gallager, “A generalized processor sharing approach to flow control in integrated service networks”, in Proceedings of IEEE INFOCOM’93, p. 521-530, 1993.
- [33] H. Tassar, H. Alnuweiri, “The complexity of computing Virtual Time in Weighted Fair Queuing Schedulers”, in IEEE International Conference of Communications, vol. 4, June 2004.
- [34] Q. Zhao, J. Xu, “On the computational complexity of maintaining GPS Clock in packet scheduling”, in Proceedings of IEEE INFOCOM’04, vol.4, March 2004.
- [35] A. K. Parekh, R. G. Gallager, “A generalized processor sharing approach to flow control in integrated service networks”, in Proceedings of IEEE INFOCOM’92, p. 915-924, 1992.
- [36] D. Nace, M. Pióro, “Max-Min Fairness and Its Applications to Routing and Load-Balancing in Communication Networks: A Tutorial”, in IEEE Communications on Surveys & Tutorials, vol. 10, no. 4, 2008.
- [37] J. Banks, J. Carson, B. L. Nelson, D. Nicol, “Discrete-event system simulation”, Prentice-Hall, 4o ed, August 2004.

[38] E. Russel, “Building Simulation Models with SIMSCRIPT II.5”, CACI Products, February 1999.

[39] S. W. Smith, “The Scientist and Engineer’s Guide to Digital Signal Processing”, California Technical Pub, 1 ed, 1997.

[40] B. Briscoe, “Flow Rate Fairness: Dismantling a religion”, in Proceedings of ACM SIGCOMM’07, Vol. 37, no.2, April 2007.

[41] D. Knuth, “The Art of Computer Programming, Vol. 1: Fundamental Algorithms”, Addison-Wesley, 3 ed, 1997.

[42] A. G. Greenberg, N. Madras, “How fair is fair queuing?”, in Journal of the ACM (JACM), Vol. 39, no. 3, July 1992.

[43] M. Gupta, M. Parihar, P. Lasalle, R. Scrimger, “TCP/IP, A Bíblia”, Elsevier, 1 ed., 2002.

8 Apêndice

8.1. Fluid-Flow Fair Queuing (GPS-Generalized Processor Sharing)

A disciplina “Fluid-Flow Fair Queueing” [35], parte do pressuposto de que o tráfego tem natureza fluida e não de pacotes, ou seja, o tráfego é um fluxo contínuo de entidades infinitesimais e não um fluxo de entidades divisíveis (pacotes).

O escalonador serve uma quantidade infinitesimal de cada fluxo (filas não-vazias). Se as filas tiverem pesos diferentes para escolha de serviço, o escalonador servirá quantidades infinitesimais proporcionalmente aos respectivos pesos. Então, esse algoritmo provê justiça aos fluxos de pacotes baseado no conceito de “Max-Min Fairness” [36].

A seguir a definição formal do algoritmo é descrita, onde se inicia com algumas definições.

Considere um sistema da seguinte forma:

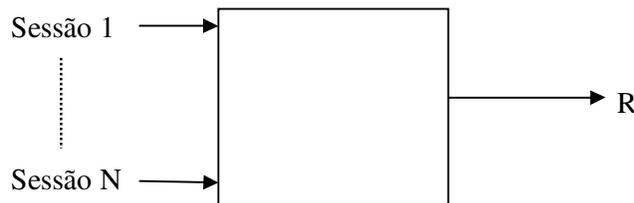


Figura 46 - Sistema de referência GPS

Onde:

ρ_i = taxa alocada a sessão i ;

$A_i(t_1, t_2)$ = taxa de chegada de pacotes da sessão i em $[t_1, t_2]$;

$W_i(t_1, t_2)$ = quantidade de tráfego servido da sessão i em $[t_1, t_2]$;

$Q_i(t_2) = A_i(t_1, t_2) - W_i(t_1, t_2) =$ tamanho da fila da sessão i em $[t_1, t_2]$;

Uma sessão é considerada como ativa no sistema se $Q_i(t_2) > 0$.

O serviço normalizado recebido pela sessão i é definido por $\frac{W_i(t_1, t_2)}{\rho_i}$;

$B(t_2)$ é o conjunto das sessões ativas;

$B(t_1, t_2)$ é o conjunto de sessões ativas no intervalo $[t_1, t_2]$;

Logo, a disciplina “Fluid-Flow Fair Queueing” se caracteriza por manter constante a seguinte razão para todos os fluxos de pacotes das sessões com filas não-vazias entre $[t_1, t_2]$:

$$\frac{w_i(t_1, t_2)}{\rho_i} = \frac{w_j(t_1, t_2)}{\rho_j}, p/ i, j \in B(t_1, t_2)$$

Nesse caso o peso considerado para cada sessão é sua taxa alocada, mas podem-se utilizar outros valores como a utilidade, instante de expiração dos pacotes, atraso medido, dentre outros.

Essa disciplina consegue associar justiça, baixa complexidade de implementação e garantia no limite do atraso, propriedades que as disciplinas tentam alcançar sempre encontrando um compromisso entre tais propriedades. Por isso que essa disciplina é utilizada como referência para outras disciplinas do tipo “Sorted-Priority”.

Além das propriedades acima listadas, o “Fluid-Flow Fair Queueing” oferece os menores atrasos, jitter e maiores taxas de transmissão por fluxo. É com base nessas características que a próxima disciplina estabelecerá sua regra de escalonamento.

No algoritmo “Fluid-Flow Fair Queueing”, o escalonador serve uma quantidade infinitesimal de cada fluxo contínuo/fluido por vez. Mas o tráfego não pode ser tratado com característica de fluido, e sim com característica de pacotes. Logo tratando o tráfego com característica de pacotes, o mecanismo do “Fluid-Flow Fair Queueing” se assemelha a servir cada bit de um pacote de cada fluxo por vez. Essa operação é a descrita pelo “Round Robin bit-a-bit”, porém esta é

uma operação não-realizável na medida em que os pacotes são indivisíveis. Então a versão em pacotes do “Fluid-Flow Fair Queueing” foi desenvolvida chamando-se “P-GPS (Packetized Generalized Processor Sharing)” [35] também conhecida como “Weighted Fair Queuing” [23].

O que a disciplina “Weighted Fair Queuing” realiza então é, calcular o instante de tempo em que o último bit de cada pacote é escalonado em uma simulação do sistema usando a disciplina “Fluid-Flow Fair Queueing”. Esse instante é chamado de “Instante Virtual de fim de serviço” (Virtual Finish Time) [33]. Logo a disciplina “Weighted Fair Queuing” escalona os pacotes de acordo com a ordem crescente dos “Instantes Virtuais de fim de serviço”.

Essa disciplina determina o tempo de vida de cada pacote baseado na atual simulação do sistema com “Weighted Fair Queueing” e o tempo de vida ideal/virtual de cada pacote baseado na simulação do sistema com “Fluid-Flow Fair Queueing”. Com o tempo de vida dos pacotes, subentende-se o instante de tempo em que o pacote termina de receber serviço (processado) menos o instante de tempo em que o pacote começa a receber serviço. Ao instante de tempo em que começa a receber serviço subentende-se não necessariamente o instante em que o pacote chega ao sistema, mas sim o instante em que é escolhido pelo escalonador para receber serviço, podendo coincidir com o instante de chegada ao sistema ou não. Já o tempo de vida calculado pelo “Fluid-Flow Fair Queueing” (considerando fluxo de bits) é utilizado para determinar os “Instantes Virtuais de fim de serviço” e consecutivamente escalonar os pacotes baseado na ordem crescente de tais instantes. A seguir segue um exemplo para entendimento desse processo.

Considerando o seguinte sistema:

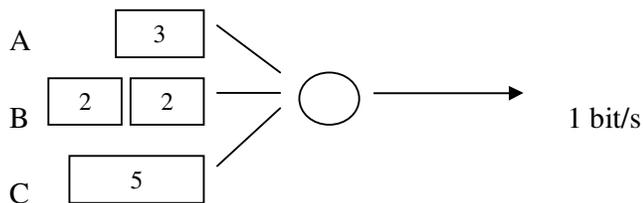


Figura 47 - Exemplo “Instante Virtual de fim de serviço”

Apenas 3 fluxos, A,B e C, competindo por serviço e um enlace de saída de 1 bit/s. Os tamanho dos pacotes de cada fluxo em bits está mostrado na figura 23 acima.

No instante zero, começo da simulação do sistema, todos os fluxos têm pacotes em suas filas, logo os instantes virtuais de fim de serviço de cada pacote são calculados da seguinte maneira:

Instante virtual de fim de serviço do pacote “p” é igual ao instante em que o escalonador se torna ocioso mais o tempo de processamento do pacote em serviço no escalonador.

O pacote do fluxo A têm instante virtual de fim de serviço de $0 + 3 = 3$, o pacote do fluxo B de $0 + 2 = 2$ e o pacote do fluxo C de $0 + 5 = 5$. Como o instante virtual de fim de serviço do pacote B é menor, ele é escalonado. Quando o segundo pacote de B chega ao sistema o seu instante virtual de fim de serviço é considerado a partir do momento em que o escalonador termina o processamento do pacote em serviço, ou seja, a partir do instante 2. Então seu valor é de $2 + 2 = 4$. Depois que o pacote em serviço é processado, o escalonador analisa os “novos” instantes virtuais de fim de serviço, $A = 3$, $B = 4$, $C = 5$ e concluiu que o pacote de A seja escalonado. No instante em que esse pacote termina seu processamento, os novos instantes virtuais são $B = 4$ e $C = 5$ e novamente o pacote de B é escalonado deixando por último o pacote de C. Essas operações são mostradas na figura 24 juntamente com sua evolução temporal.

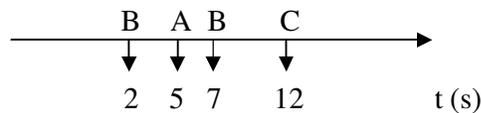


Figura 48 - Escala do tempo “Instante Virtual de fim de serviço”

Nesse exemplo, o cálculo do “Instante Virtual de fim de serviço” é um tanto trivial, mas quando várias sessões são levadas em consideração, essa tarefa se torna um tanto complicada [33,34]. Na tentativa de reduzir a complexidade nos cálculos dos “Instantes Virtuais de fim de serviço” mantendo um desempenho desejável é que a disciplina “Self-Clocked Fair Queuing” foi desenvolvida.