PACKET SCHEDULING AND PRICING BASED ON INFLICTED DELAY

Ari Viinikainen
Department of Mathematical Information Technology, University of Jyväskylä
P.O. Box 35 (Agora)
40014 University of Jyväskylä, Finland

Keywords: Pricing, revenue optimization, delay, Quality of Service (QoS).

Abstract: An adaptive packet scheduling method is presented in this paper. The adaptive weights of a scheduler are chosen based on maximizing the revenue of the network service provider. The pricing scenario is based on the delay that a connection will inflict to other connections. The features of the adaptive weight updating algorithm are simulated, analyzed and compared to a constant weight algorithm.

1 INTRODUCTION

Internet services should be focused more on service alignment and revenue generation. This requires new ideas for network management solutions. To meet growing organizational demands, policy-based management is one solution that will allow organizations to prioritize networking resources such as bandwidth, application access and security clearance based on individual users. These policy-based management methods will have to be self-deploying, self-configuring and self-healing, automatically discovering any changes taking place in the network infrastructure and dynamically building and altering policies for accessing resources based on needs. By combining scheduling and revenue issues together an efficient method can be built to allocate network resources in optimal and profitable way.

The packet scheduling algorithms can be classified into two categories, depending on the need for sorting the packets in the scheduler. Those that are based on packet sorting, need to maintain a “virtual time”, that is used to calculate the order of the packets. These kinds of schedulers provide good fairness and low latency but due to the complexity in computation of the parameters used for sorting, they lack efficiency. Weighted Fair Queueing (WFQ) (Demers et al., 1989; Parekh and Gallager, 1993) was a first proposal of these kinds. Later other proposals as Self-clocked Fair Queueing (SCFQ) (Golestani, 1994), Worst-case Fair Weighted Queueing (WF^2Q) (Bennett and Zhang, 1996), Start-time Fair Queueing (SFQ) (Goyal et al., 1997) and Frame-Based Fair Queueing (FFQ) (Stiliadis and Varma, 1998) have been proposed to combat the lack of efficiency.

Schedulers that handle packets in a round robin manner, like Deficit Round Robin (DRR) (Shreedhar and Varghese, 1996), are in the other category. These have low complexity but they are not as good in fairness and latency. Several algorithms have been proposed for improving the latency and fairness in a round robin based schemes, such as Advanced Round Robin (ARR) (Marosits et al., 2001), Smoothed Round Robin (SRR) (Guo, 2001), Pre-order deficit round robin (Tsao and Lin, 2001), Nested DRR (Kanhere and Sethu, 2001), Elastic Round Robin (ERR) (Kanhere et al., 2002), Stratified Round Robin (Ramabhadran and Pasquale, 2003) and Fair Round Robin (FRR) (Yuan and Duan, 2005).

This paper proposes a scheduling model that guarantees that the latencies for different service classes are appropriate and at the same time optimizes the network service provider’s revenue. This work continues the work in (Viinikainen et al., 2004) and (Joutsensalo et al., 2005) where the pricing of a connection was based on delay and bandwidth that the connection itself obtains, respectively. The proposed algorithm that is presented in this paper ensures less delay for the users paying more for the connection (i.e. higher service class) than those paying less. The proposed method also penalizes by pricing those customers, who induce the most delay in the network. Thus, this approach yields a tempting scenario for customers and the service provider. The closed form

Viinikainen A. (2006).
PACKET SCHEDULING AND PRICING BASED ON INFLICTED DELAY.
In *Proceedings of the International Conference on e-Business*, pages 113-117
DOI: 10.5220/0001424201130117

formula for updating weights is independent on any statistical behavior of the connections. Therefore, it is also robust against erroneous estimates of customers' behavior.

The rest of the paper is organized as follows. First, in Section 2 the scheduler is discussed and the proposed pricing scenario is defined, whilst experiments justifying the analytical derivation are made in Section 3. Section 4 contains discussion of theory and experiments and Section 5 concludes the study.

2 THE PACKET SCHEDULER AND DELAY

Let us consider a packet scheduler which receives packets to be delivered from m different queues (i.e. classes). Now, let T be the processing time of the classifier for transmitting a data packet from one queue to the output of a packet scheduler. For any connection k there are $N_{ik}(t)$ packets in the queue of class i at some time t. Total number of packets in the queue i at the time t is therefore

$$N_i(t) = \sum_{k=1}^{K_i(t)} N_{ik}(t), \tag{1}$$

where $K_i(t)$ is the total number of connections in the class i at time t.

Now, every packet in the queue increases delay for the other connections in the queue so a new connection k appearing at time t in class i will increase the delay to the other connections by a value proportional to $N_{ik}(t)$

$$d_{ik}(t) = \frac{N_{ik}(t)T}{w_i(t)} = \frac{N_{ik}(t)}{w_i(t)}, \tag{2}$$

where T can be scaled to $T = 1$ without loss of generality and $w_i(t) = w_i, i = 1, \ldots, m$ are weights allocated for each class as a new connection in any class will inflict delay on the other classes depending on amount of processing time of the scheduler allotted to each class. The overall delay in class i is then

$$D_i(t) = \sum_{k=1}^{K_i} d_{ik} = \sum_{k=1}^{K_i} \frac{N_{ik}}{w_i}, \tag{3}$$

where the time index t has been dropped for convenience until otherwise stated. The natural constraints for the weights are

$$w_i > 0 \tag{4}$$

and

$$\sum_{i=1}^{m} w_i = 1. \tag{5}$$

For every connection the pricing is based on the relative increase in delay that it produces. Every connection induce delay and those connections that induce more delay than the connections on average will pay more for the connection.

Definition 1. For each service class the function

$$P_i = \sum_{k=1}^{K_i} K_i r_i + s_i d_{ik}^p, r_i > 0, s_i > 0, p > 0 \tag{6}$$

is called a polynomial pricing function.

In the polynomial pricing function r_i is a base price of a connection in the class i, s_i is a penalty factor that increases the price for those connections which induce more delay and p assures that the price does not grow too rapidly with the delay. The total revenue of the operator is

$$R = \sum_{i=1}^{m} P_i = \sum_{i=1}^{m}\sum_{k=1}^{K_i} p_{ik} = \sum_{i=1}^{m}\sum_{k=1}^{K_i} K_i r_i + s_i d_{ik}^p. \tag{7}$$

Theorem 1. *Optimal weights for the revenue R under the polynomial pricing model is*

$$w_i = \frac{s_i^{\frac{1}{p-1}} N_i^{\frac{p}{p-1}}}{\sum_{j=1}^m s_j^{\frac{1}{p-1}} N_j^{\frac{p}{p-1}}}. \tag{8}$$

when $0 < p < 1$.

Proof. By using Lagrangian approach, the revenue can be presented in the form

$$R = \sum_{i=1}^{m}\sum_{k=1}^{K_i} K_i r_i + s_i \frac{N_{ik}^p}{w_i^p} + \lambda(1 - \sum_{i=1}^{m} w_i). \tag{9}$$

Optimal weights are obtained from the first derivative

$$\frac{\partial R}{\partial w_i} = \sum_{k=1}^{K_i} -p s_i \frac{N_{ik}^p}{w_i^{p-1}} - \lambda = 0. \tag{10}$$

Because Lagrangian solution

$$\frac{\partial F}{\partial \lambda} = 0 \tag{11}$$

yields $\sum_{j=1}^m w_j = 1$, then

$$\begin{aligned} w_i &= \left(\frac{-p s_i N_i^p}{\lambda}\right)^{\frac{1}{p-1}} \\ &= \frac{(-p s_i)^{\frac{1}{p-1}} N_i^{\frac{p}{p-1}}}{\lambda^{\frac{1}{p-1}} \sum_{j=1}^m w_j} \\ &= \frac{(-p s_i)^{\frac{1}{p-1}} N_i^{\frac{p}{p-1}}}{\lambda^{\frac{1}{p-1}} \sum_{j=1}^m \frac{(-p s_j)^{\frac{1}{p-1}} N_j^{\frac{p}{p-1}}}{\lambda^{\frac{1}{p-1}}}} \\ &= \frac{s_i^{\frac{1}{p-1}} N_i^{\frac{p}{p-1}}}{\sum_{j=1}^m s_j^{\frac{1}{p-1}} N_j^{\frac{p}{p-1}}}. \end{aligned} \tag{12}$$

From the expression (12) it is seen that λ has the form

$$\lambda^{\frac{1}{p-1}} = \sum_{j=1}^{m} s_j^{\frac{1}{p-1}} N_j^{\frac{p}{p-1}}, \tag{13}$$

or

$$\lambda = \sum_{j=1}^{m} s_j N_j^p. \tag{14}$$

Thus the first order derivative is

$$\frac{\partial R}{\partial w_i} = \sum_{k=1}^{K_i} \frac{-p s_i N_{ik}^p}{w_i^{p-1}} - \sum_{j=1}^{m} s_j N_j^p, \tag{15}$$

and so the second order derivative is

$$\frac{\partial^2 R}{\partial w_i^2} = \sum_{k=1}^{K_i} -(p-1)\frac{-p s_i N_{ik}^p}{w_i^{p-2}} < 0 \tag{16}$$

when $0 < p < 1$. In that condition, R is concave with respect to the weights w_i, and the optimal solution is unique. □

By using optimal weights (8), revenue R can be expressed as follows:

$$\begin{aligned} R &= \sum_{i=1}^{m} \sum_{k=1}^{K_i} K_i r_i + \frac{s_i N_i}{w_i} \\ &= \sum_{i=1}^{m} K_i r_i + \frac{s_i N_i}{\frac{s_i^{\frac{1}{p-1}} N_i^{\frac{p}{p-1}}}{\sum_{j=1}^{m} s_j^{\frac{1}{p-1}} N_j^{\frac{p}{p-1}}}} \\ &= \sum_{i=1}^{m} K_i r_i + s_i^{\frac{p-2}{p-1}} N_i^{\frac{-1}{p-1}} \\ &\times \sum_{j=1}^{m} s_j^{\frac{1}{p-1}} N_j^{\frac{p}{p-1}}. \end{aligned} \tag{17}$$

3 EXPERIMENTS

In this section the operation of the pricing algorithm (6) with $p = \frac{1}{2}$ and $m = 3$ service classes is demonstrated by simulation. The optimal weights (12) are compared with a constant weight ($w_1 = 1/2, w_2 = 1/3$ and $w_3 = 1/6$) version of the algorithm.

The processing time of the server is chosen as $T = 1/1000$ s/packet. The life time of a connection is exponentially distributed. The arrival rates of the connections are Poisson distributed and they are $\alpha_1 = 0.30$, $\alpha_2 = 0.40$ and $\alpha_3 = 0.50$ per unit time for the gold, silver, and bronze classes, respectively. The duration parameters (i.e. "decay rates") for the connections are $\beta_1 = 0.015$, $\beta_2 = 0.013$ and $\beta_3 = 0.011$, where the probability density functions for the durations are

$$p_i(t) = \beta_i e^{-\beta_i t}, \quad , i = 1, 2, 3 \quad t \geq 0. \tag{18}$$

The number of unit times in the experiments was $\tau = 2000$ which is 2 seconds using T as defined above. In the experiments the pricing factors $r_i = 0$ as they only make a shift in the revenue curves (i.e. increase the revenues). The penalty factors were chosen as $s_1 = 10$, $s_2 = 15$ and $s_3 = 20$ for gold, silver and bronze classes, respectively. So connections causing delay in the gold class are penalized less and those in bronze class the most.

The number of connections on the average is highest for the bronze class and lowest for the gold class as is seen in Fig. 1. The number of connections also varies greatly over time. Each connection has constant value of 1 to 5 packets in the queue during their life time. The delays for the services classes when using constant weights are shown in Fig. 2. It is observed that the delays remain quite constant because the weights are constant. By using adaptive weights a delay critical class used for e.g. real time traffic can be favored with respect to delay. In the experiment the parameters s_i were chosen so that with adaptive weights the gold class customers get less delay with the expense of the other classes (Fig. 4). By adjusting the penalty factors s_i the relative delays between the classes can be adjusted. The values of mean delays

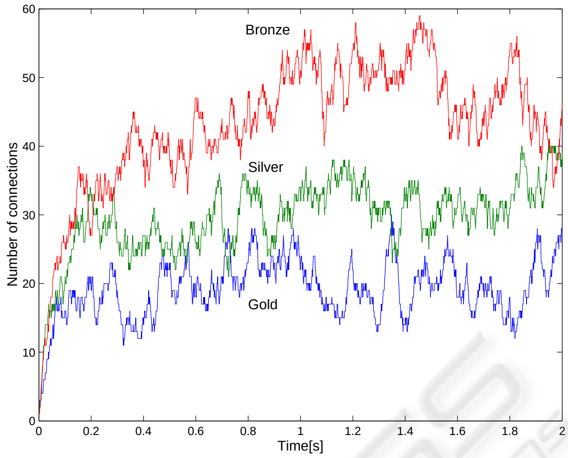

Figure 1: Evolution of the number of connections as a function of time.

Table 1: Mean delays of gold, silver and bronze class for the constant and adaptive weight algorithms.

Mean delays [s]	Gold	Silver	Bronze
Constant weights	0.1133	0.2642	0.7712
Adaptive weights	0.0804	0.4475	1.7155

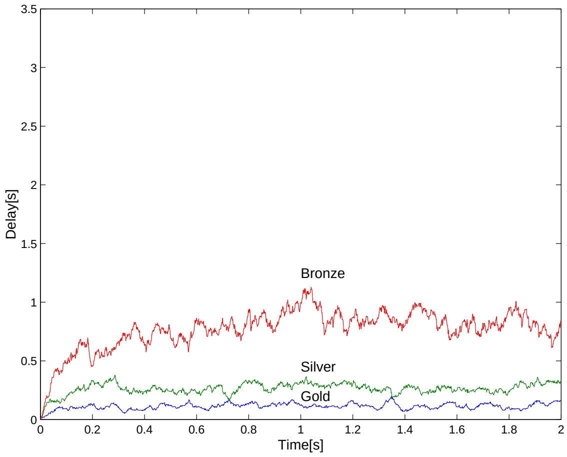

Figure 2: Constant weights - Evolution of the delay as a function of time.

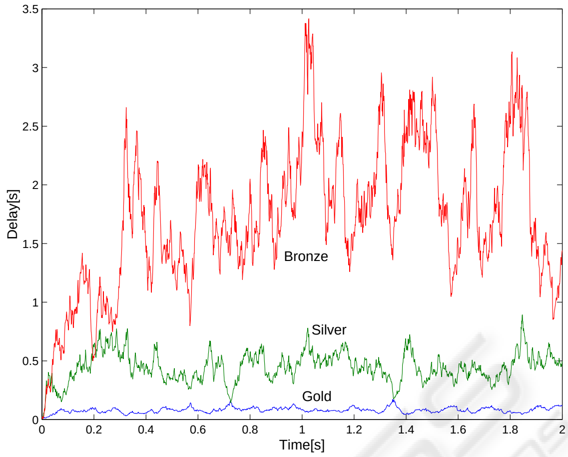

Figure 4: Adaptive weights - Evolution of the delay as a function of time.

for the classes in both cases are seen In Table 1.

The adaptive weights are seen in Fig. 3 and the total revenues for both versions of the algorithm is seen in Fig. 5. Mean revenues in case of constant weights was 5273.9 and in the case of optimal adaptive weights 7297.0. The adaptive weights give greater total revenue and smaller delay for the gold class.

4 DISCUSSION

In this section, we discuss the algorithm from the point of view of theory and experiments. Analytic forms of the revenue and the weights, that allocate traffic to the connections of different service classes were defined. The updating procedure is deterministic and nonparametric i.e. it does not make any assumptions of the statistical behavior of the traffic and connections. Thus it is robust against the errors that may occur from erroneous models. The algorithm is unique and optimal, which has been theoretically proved by Lagrangian optimization method. Closed form solution makes the algorithm quite simple. By changing the penalty factor s_i of one class higher, the delays in other classes are decreased. Because all penalty factors s_i are positive, all classes obtain service in fair way. The method always optimizes the operator's revenue and by knowing what kind of traffic is in different service classes the parameters can be set so that the delay for delay critical traffic can be maintained low enough. A call admission control mechanism could also be implemented to assure def-

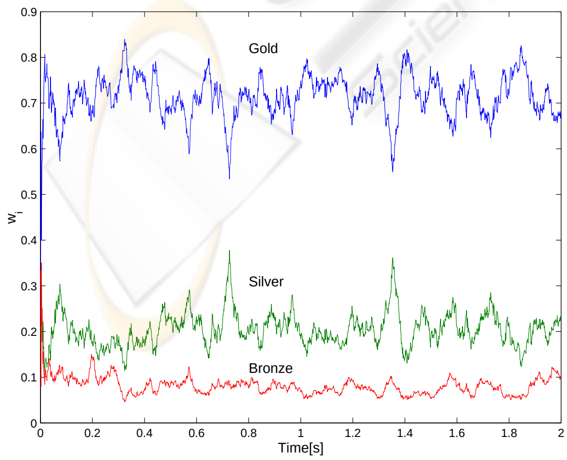

Figure 3: Evolution of the weights w as a function of time.

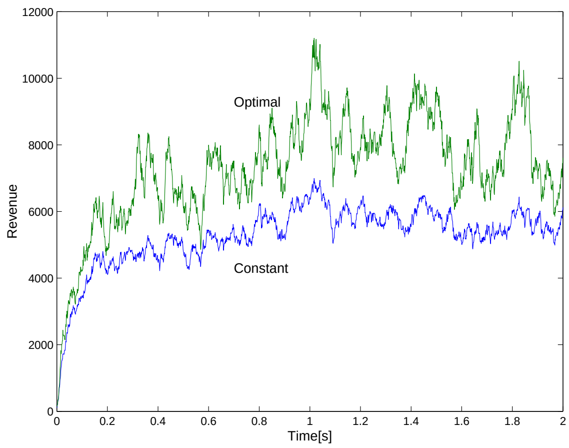

Figure 5: Evolution of the revenue R as a function of time.

inite delay limits in cases the network load increases too much. As the delays grow the operators revenue increases, but it cannot increase to infinity in reality. This is because when the buffers get full and packets start to drop, connections drop also at some point. A penalty term for the operator is under study, which would lower the price when there is too much delay.

5 CONCLUSION

In this paper an adaptive algorithm for optimizing the network operator revenue and ensuring delay as a Quality of Service (QoS) requirement was presented. The closed form algorithm was derived from a revenue-based optimization problem. The pricing of the connection is based on the delay that it inflicts on other connections. The customers that induce most delay to the network by large amounts of traffic are charged the most. In the experiments we simulated the operation of the adaptive algorithm and compared it with a constant weight version of the same algorithm. The obtained results show that by using the adaptive weights the delays can be adjusted so that the users is different QoS classes are content. With the constant weights, the delays cannot be controlled. Also, the revenue is maximized while ensuring that the customers delays stay small according to the their class. The algorithm is deterministic and non-parametric, and thus in practical environments it is a competitive candidate due to its robustness. Future work shall concentrate on expanding the idea to several nodes with more QoS parameters.

REFERENCES

Bennett, J. C. R. and Zhang, H. (1996). WF^2Q: worst-case fair weighted fair queueing. In *Proc. 15th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'96)*, volume 1, pages 120 – 128.

Demers, A., Keshav, S., and Shenker, S. (1989). Analysis and simulation of a fair queuing algorithm. *ACM SIGCOMM Comput. Commun. Rev.*, 19(4):1–12.

Golestani, S. J. (1994). A self-clocked fair queueing scheme for broadband applications. In *Proc. 13th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'94)*, volume 2, pages 636 – 646.

Goyal, P., Vin, H. M., and Cheng, H. (1997). Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Trans. Networking*, 5(5):690–704.

Guo, C. (2001). SRR: an $O(1)$ time complexity packet scheduler for flows in multi-service packet networks. *ACM SIGCOMM Comput. Commun. Rev.*, 31(4):211–222.

Joutsensalo, J., Viinikainen, A., Hämäläinen, T., and Wikström, M. (2005). Bandwidth allocation and pricing for telecommunications network. In *Proc. Next Generation Internet Networks (NGI'05)*, pages 165 – 172.

Kanhere, S. S. and Sethu, H. (2001). Fair, efficient and low-latency packet scheduling using nested deficit round robin. In *Proc. IEEE Workshop on High Performance Switching and Routing*, pages 6 – 10.

Kanhere, S. S., Sethu, H., and Parekh, A. B. (2002). Fair and efficient packet scheduling using elastic round robin. *IEEE Trans. Parallel Distrib. Syst.*, 13(3):324–336.

Marosits, T., Molnár, S., and Sztrik, J. (2001). CAC algorithm based on advanced round robin method for QoS networks. In *Proc. Sixth IEEE Symposium on Computers and Communications, 2001*, volume 2, pages 266 – 274.

Parekh, A. K. and Gallager, R. G. (1993). A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Trans. Networking*, 1(3):344–357.

Ramabhadran, S. and Pasquale, J. (2003). Stratified round robin: A low complexity packet scheduler with bandwidth fairness and bounded delay. In *Proc. 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 239 – 250. ACM Press.

Shreedhar, M. and Varghese, G. (1996). Efficient fair queuing using deficit round-robin. *IEEE/ACM Trans. Networking*, 4(3):375–385.

Stiliadis, D. and Varma, A. (1998). Efficient fair queueing algorithms for packet-switched networks. *IEEE/ACM Trans. Networking*, 6(2):175–185.

Tsao, S.-C. and Lin, Y.-D. (2001). Pre-order deficit round robin: a new scheduling algorithm for packet-switched networks. *Comput. Networks*, 35(2-3):287–305.

Viinikainen, A., Joutsensalo, J., Pääkkönen, M., and Hämäläinen, T. (2004). Packet scheduling algorithm with weight optimization. In *Proc. International Conference on E-business and Telecommunication Networks (ICETE'04)*, volume 2, pages 127 – 133.

Yuan, X. and Duan, Z. (2005). FRR: a proportional and worst-case fair round robin scheduler. In *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, volume 2, pages 831– 842.