The Closed Form Algorithm for QoS Supported Scheduling

Jyrki Joutsensalo[1],
Timo Hämäläinen[1], and
Gabriele Peters[2]
[1]Department of Mathematical Information Technology
University of Jyväskylä
40014 Jyväskylä, FINLAND
Emails: timoh@cc.jyu.fi, jyrkij@mit.jyu.fi
[2]Computer Science Visual Computing
University of Applied Sciences and Arts
Emil-Figge-Strae 42, D-44227 Dortmund, GERMANY
Email: gabriele.peters@fh-dortmund.de

I. Abstract

This paper presents a dynamic scheduling algorithm. The purpose of the algorithm is to maximize revenue of the network service provider and share network resources at the fair way. Presented algorithm is derived from the linear type of revenue target function, and closed form globally optimal formula is presented. The method is computationally inexpensive, while still producing maximal revenue. Due to the simplicity of the algorithm, it can operate in the highly non-stationary environments. In addition, it is non-parametric and deterministic in the sense that it uses only the information about the number of users and their traffic classes, not about call density functions or duration distributions.

II. Introduction

Today's networks must carry a wide range of different traffic types being still able to provide performance guarantees to realtime traffic such as Voice over IP (VoIP), Video-on-Demand (VoD), or videoconferencing and at the same time give some capacity to the best effort traffic. At the point of single node view the traffic classification is handled by scheduling multiple queues in a matter that leads to the optimal result. Most popular scheduling algorithms are priority queue [8] and weighted fair queue (WFQ) [9]. Priority queue prefers classes with higher priority, but it is nonadaptive and unfair, i.e. the delay in the low priority queue may increase unreasonably large. In contrast, WFQ gives weights for different classes in such a way that the performance of the low priority queues is guaranteed.

Traffic classification and pricing of the services are the issues we need to combine. Many optimal pricing schemes have been proposed to address this problem. Most of approaches assume an known user utility function and establish a optimization model to either maximize the user benefit or provider revenue [6], [7], [10]. However, the major problem with this kind of approach is that user utility function can not be well defined in short term and sometimes even very difficult in long term. The effectiveness of such schemes is still questionable. This paper extends our previous pricing and QoS research, in which the optimal link allocation between traffic classes using different pricing scenarios and the QoS were studied [2]. The possibility of using revenue as the criterion for updating weights in the WFQ service discipline case was theoretically considered in linear and flat pricing scenario [3], [5]. In this paper we take into account queuing scheduling issues by introducing dynamic weight tracking algorithm in the scheduler. QoS and revenue aware scheduling algorithm is investigated in the single node case. It is derived from Lagrangian optimization problem, and globally optimal closed form solution is presented. The close research is [1], where adaptive WFQ algorithm was investigated, but revenue criterion was not used.

III. Pricing scenario

Here the pricing scenario is presented in the simplified form. Let d_0 be the minimum processing time of the classifier for transmitting data from one queue to the output. For simplicity it is assumed that the data packets have the same size b. Therefore their size can be scaled to $b = 1$. Extensions to the variable packet sizes do not need essential modifications to the main theory. The number of service classes is denoted by m, and in this case, $m = 3$. In each queue, sub-queues can be defined due to the different insertion delays, transmission delays etc. of the different packets in the same queue. However, this is also straightforward extension to our scenario, and therefore it is beyond the scope of this study. It has only the effect on the computational complexity. In our scheduling model, real processing time (delay) for class i in the packet scheduler is $d_i = N_i d_0 / w_i$, where $w_i(t) = w_i, i = 1, \ldots, m$ are weights allotted for each class, and $N_i(t) = N_i$ is a number of customers in the ith queue. Here time index t has been dropped for convenience. The natural constraints for the weights are

$$w_i > 0 \tag{1}$$

and

$$\sum_{i=1}^{m} w_i = 1. \tag{2}$$

Without loss of generality, only non-empty queues are considered, and therefore $w_i \neq 0, i = 1, \ldots, m$. If some weight is $w_i = 1$, then $m = 1$, and the only class to be served has the minimum processing time d_0, if $N_i = 1$. For each service class, a revenue or *pricing function*

$$r_i(d_i) = r_i(N_i d_0/w_i + c_i) \tag{3}$$

(euros/minute) is non-increasing with respect to the delay d_i. Here $c_i(t) = c_i$ includes insertion delay, transmission delay etc., and here it is assumed to be constant (therefore above-mentioned sub-queue systems are not considered here). In this paper our study concentrates to the case of the simplest functions, namely linear pricing functions, as shown in Fig. 1. Linear pricing algorithms may perhaps also be used as building blocks for developing piecewise linear pricing models.

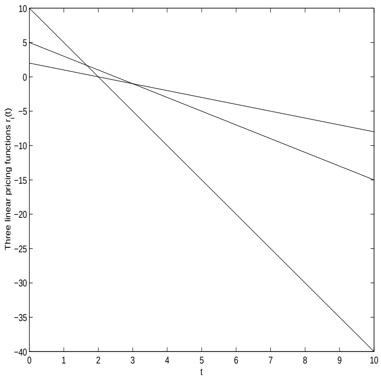

Fig. 1. Three linear pricing functions. Horizontal axis: delay; vertical axis: price.

For gold class, the pricing model $r_1(d) = -5d + 10$ means that when the delay d is small, the price paid by gold class customer is high - maximally 10 units of money. It is natural that for the highest priority class, constant shift (e.g. ten money units in this case) is selected to be highest. On the other hand, penalty paid to the highest priority class customers is also highest; in this case it depends linearly on the delay, being $-5d$. For example, if $d = 3$, then $r_1(d) = r_1(3) = -5 \times 3 + 10 = -5$ units of money. Same observations hold for silver and bronze classes. For bronze class, $r_3(d) = -d + 2$ means that the price paid by that class customer is maximally 2 units of money. In this case, constant shift was selected to be lowest. On the other hand, penalty for bronze class is also lowest, being $-d$. However, our purpose is not to make accurate study of the practical realizations of the parameters of the curves, only general parametrical forms of the pricing functions.

IV. Scheduling algorithm

One user in class i pays $r_i(d_i)$ money units to the service provider according to the pricing function (3). Because there are N_i customers in the queue i, the total price paid by the ith class customers in unit time step (euros/minute) is $N_i r_i(N_i d_0/w_i + c_i)$. Because there are m classes, the revenue criterion to be maximized has the form

$$F(w_1, \ldots, w_m) = \sum_{i=1}^{m} N_i r_i(N_i d_0/w_i + c_i) \tag{4}$$

under weight constraint (1) and (2). Without loss of generality, set $d_0 = 1$.

As a special case, consider *linear* revenue model.

Definition 1: *The function*

$$r_i(t) = -r_i t + k_i, \quad i = 1, \ldots, m, \tag{5}$$

$$r_i > 0, \tag{6}$$

$$k_i > 0, \tag{7}$$

is called linear pricing function.

Using Eqs. (3), (4) and (5), we define the revenue F for linear pricing functions by Lagrangian as follows:

$$\begin{aligned} F &= F(w_1, \ldots, w_m) \\ &= \sum_{i=1}^{m} N_i(-r_i \frac{N_i}{w_i} + k_i) + \lambda(1 - \sum_{i=1}^{m} w_i) \\ &= -\sum_{i=1}^{m} \frac{r_i N_i^2}{w_i} + \sum_{i=1}^{m} N_i k_i + \lambda(1 - \sum_{i=1}^{m} w_i), \end{aligned}$$

$$0 < w_i \leq 1. \tag{8}$$

Here the constants c_i have been dropped out for convenience. Theorem of closed form solution for optimal weights is as follows:

Theorem 1: *For linear pricing functions, the maximum revenue F is achieved by using the weights*

$$w_i = \frac{\sqrt{r_i} N_i}{\sum_{l=1}^{m} \sqrt{r_l} N_l}, \tag{9}$$

and it is unique in $w_i \in (0, 1]$. We have proofed that in [**?**], and an upper bound for revenue F is obtained:

Theorem 2:

$$F < \sum_{i=1}^{m} N_i k_i. \tag{10}$$

This is proofed in [4]. Analytical form to the revenue can be expressed solving weights w_i out:

Theorem 3: *When optimal weights* w_i *are used according to Theorem 1, revenue is*

$$F = -\left(\sum_{i=1}^{m} \sqrt{r_i} N_i\right)^2 + \sum_{i=1}^{m} N_i k_i. \tag{11}$$

Proof: When penalty $\lambda(1 - \sum_i w_i)$ in Eq. (8) vanishes, F can be represented in the form

$$F = -\sum_{i=1}^{m} \frac{r_i N_i^2}{w_i} + \sum_{i=1}^{m} N_i k_i. \tag{12}$$

Substitute optimal weights (9) to Eq. (12). Then

$$\begin{aligned} F &= -\sum_{i=1}^{m} r_i N_i^2 \frac{\sum_{l=1}^{m} \sqrt{r_l} N_l}{\sqrt{r_i} N_i} + \sum_{i=1}^{m} N_i k_i \\ &= -\sum_{i=1}^{m} \sqrt{r_i} N_i \sum_{l=1}^{m} \sqrt{r_l} N_l + \sum_{i=1}^{m} N_i k_i \\ &= -(\sum_{i=1}^{m} \sqrt{r_i} N_i)^2 + \sum_{i=1}^{m} N_i k_i. \end{aligned} \tag{13}$$

Q.E.D.

From Eq. (13), one possible constraint in the CAC mechanism is obtained, namely

$$(\sum_{i=1}^{m} \sqrt{r_i} N_i)^2 < \sum_{i=1}^{m} N_i k_i, \tag{14}$$

that guarantees $F > 0$.

Next theorem states optimal number of users, as well as upper bounds for buffer sizes:

Theorem 4:*Upper bounds for buffer sizes are*

$$q_i = \lfloor \frac{1}{2} \frac{k_i}{r_i} \rfloor, \quad i = 1, \ldots, m, \tag{15}$$

where $y = \lfloor x \rfloor$ denotes maximum integer y satisfying $y \leq x$.

Proof: The optimal number of users for fixed weights is obtained as follows:

$$\frac{\partial F}{\partial N_l} = -2 \frac{r_l}{w_l} N_l + k_l = 0. \tag{16}$$

Therefore

$$N_l = \frac{1}{2} \frac{w_l k_l}{r_l}, \quad l = 1, \ldots, m. \tag{17}$$

The second derivative is

$$\frac{\partial^2 F}{\partial N_l^2} = -2 \frac{r_l}{w_l} < 0, \tag{18}$$

because $r_l > 0$ and $w_l \geq 0$. Therefore F is strictly concave with respect to N_i, $i = 1, \ldots, m$ having one and only one global maximum, which is satisfied by Eq. (17). Because $w_i \leq 1$, $i = 1, \ldots, m$, then

$$N_l \leq \frac{1}{2} \frac{k_l}{r_l}, \tag{19}$$

for which Eq. (15) follows. This completes proof. **Q.E.D.**

Next another upper bound for revenue is presented:

Theorem 5: *In the case of linear pricing model (1), upper bound for revenue is*

$$F \leq \frac{1}{4} \sum_{i=1}^{m} \frac{k_i^2}{r_i}. \tag{20}$$

Proof: Select optimal value for N_i in Eq. (17), and substitute it in Eq. (8) by using constraint (2). Then

$$F = \sum_{i=1}^{m} \frac{1}{2} \frac{w_i k_i}{r_i} \left(-r_i \frac{1}{2} \frac{w_i k_i}{r_i w_i} + k_i \right) = \frac{1}{4} \sum_{i=1}^{m} \frac{w_i k_i^2}{r_i}. \tag{21}$$

Due to the condition $w_i \leq 1$, Eq. (20) follows. **Q.E.D.**

Interpretation of (20) is quite obvious: k_i increases upper limit, while r_i decreases it.

Call Admission Control mechanism can be made by simple hypothesis testing without assumptions about call or dropping rates. Let the state at the moment t be $N_i(t)$, $t = 1, \ldots, m$. Let the new hypothetical state at the moment $t+1$ be $\tilde{N}_i(t+1)$, $t = 1, \ldots, m$, when one or several calls appear. In hypothesis testing, Theorem 3 is applied as follows:

$$F(t) = -\left(\sum_{i=1}^{m} \sqrt{r_i} N_i(t) \right)^2 + \sum_{i=1}^{m} N_i(t) k_i. \tag{22}$$

$$\tilde{F}(t+1) = -\left(\sum_{i=1}^{m} \sqrt{r_i} \tilde{N}_i(t+1) \right)^2 + \sum_{i=1}^{m} \tilde{N}_i(t+1) k_i. \tag{23}$$

If $F(t) > \tilde{F}(t)$, then call is rejected, otherwise it is accepted.

Computational complexity of the algorithm also be derived by exploiting Theorem 3. When no calls or droppings happen, weights are not adjusted. When call appears, $O(m)$ multiplications and additions are performed, as seen from Eq. (11).

V. Experiments

In the experiments, call arrivals and duration are Poisson and exponentially distributed. Call rates per unit time for gold, silver, and bronze classes are $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, and $\alpha_3 = 0.3$, respectively. Duration parameters (decay rates) are $\beta_1 = 0.010$, $\beta_2 = 0.007$, and $\beta_3 = 0.003$, where probability density functions for duration are

$$f_i(t) = \beta_i e^{-\beta_i t}, \quad , i = 1, 2, 3, \quad t \geq 0. \tag{24}$$

The three service classes have pricing functions as follows:

$$r_1(t) = -5t + 200 \tag{25}$$

for gold class,

$$r_2(t) = -2t + 100 \tag{26}$$

for for silver class, and

$$r_3(t) = -0.5t + 50 \tag{27}$$

for bronze class. Figure 2 shows the evolution of three weights $w_1(t)$, $w_2(t)$, and $w_3(t)$ as a function of time, with and without CAC mechanism. Figure 3 shows the corresponding delays. Solid, dashed, and dash-dotted curves correspond to gold, silver, and bronze class, respectively. It is not surprising that the delays of gold class customers are lowest, while delays of bronze class customers are largest. Number of users $N_i(t)$ are shown in Fig. 4. Due to the arrival and duration rates, number of users is lowest in gold class, while number of users is largest in bronze class. Solid, dashed, and dash-dotted lines show upper bounds of the different buffers according to the Theorem 4. However, because CAC mechanism is not used (left fig.), and $N_i(t)$ may be are larger than the upper bounds. $N_i(t)$ achieves the theoretical value

$$E[N_i(t)] = \frac{\alpha_i}{\beta_i} \tag{28}$$

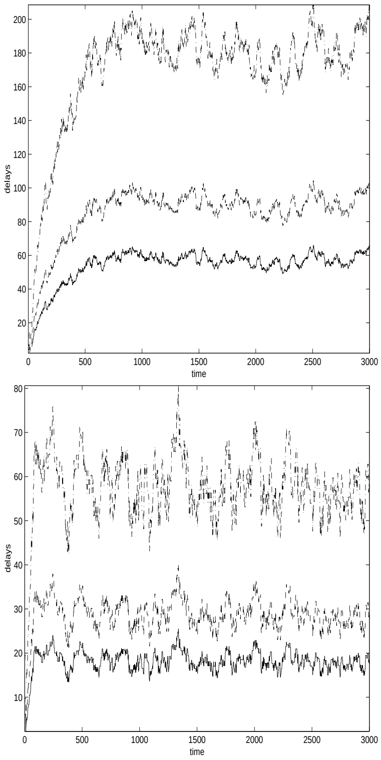

Fig. 3. Delays as a function of time. Left fig. without CAC and right fig. with CAC. Horizontal axis: time. Vertical axis: delay. Solid, dashed, and dash-dotted curves correspond to gold, silver, and bronze class, respectively.

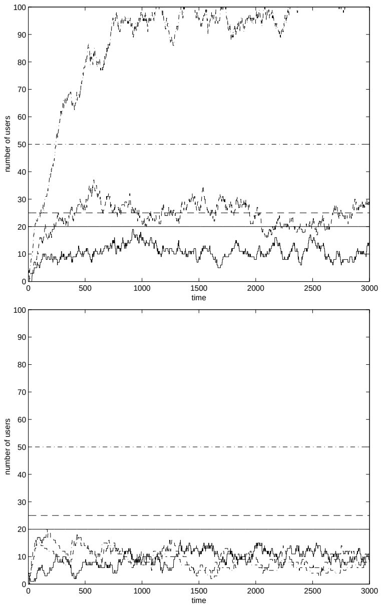

Fig. 4. Left fig. without CAC and right fig. with CAC. Horizontal axis: time. Vertical axis: number of users. Solid, dashed, and dash-dotted curves correspond to gold, silver, and bronze class, respectively. Lines are upper bounds for buffer sizes.

stated in Little's Theorem, i.e. $\alpha_1/\beta_1 = 10$, $\alpha_2/\beta_2 \approx 29$, $\alpha_3/\beta_3 = 100$. In Fig. 5, revenue as well as two upper bounds are shown (note y-axis scale is different in the subfigs). At the left fig. the lowest curve (solid) is the revenue achieved by the closed form method with no CAC. It became negative. Dashed curve shows the upper limit $\sum_i N_i k_i$ as stated in Theorem 2, and the solid line illustrates the upper bound of Theorem 5, and it is constant due to the invariance of r_i and k_i in that experiment. When using our CAC mechanism it is noticeable, that the upper bounds are larger than the realized number of users, as well as revenue, and what is important, revenue is now positive.

VI. Conclusions

Experiments clearly justify the performance of the developed dynamic scheduling algorithm. For example, theorems for upper bounds hold, and revenue curves are positive. Some of the statistical and deterministic algorithms presented in the literature assume quite strict *a priori* information about parameters or statistical behavior such as call densities, duration or distributions. However, such methods usually are - in addition to computationally complex - not robust against erroneous assumptions or estimates. On the contrary, our algorithm is deterministic and non-parametric, ie. it uses only the information about the number of customers, and thus we believe that in practical environments it is comptetitive candidate due to the robustness. Also, Call Admission Control (CAC) mechanism can be used in the context of the algorithm. It is based on the hypothesis testing, and is computationally quite simple. The algorithm used the same packet sizes. However,

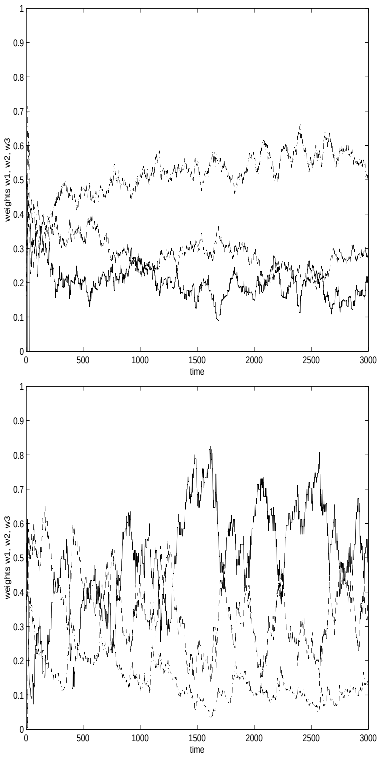

Fig. 2. Three weights as a function of time. Left fig. without CAC and right fig. with CAC. Horizontal axis: time. Vertical axis: weight value.

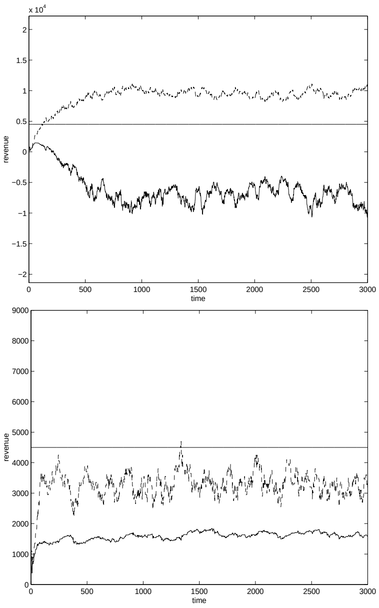

Fig. 5. Left fig. without CAC and right fig. with CAC. Horizontal axis: time. Vertical axis: revenue. Lowest curve: realized revenue; middle curve (dashed): upper bound $\sum_i N_i k_i$; solid line: $0.25 \sum_i k_i^2/r_i$.

it is quite straightforward to develop the version, which can handle different packet sizes. General conclusion is that the linear pricing scenario is quite simple. However, we believe that more practical pricing scheme can be based on *piecewise* linear model. Studies to that direction are made. Especially flat pricing scenario is interesting topic of study.

References

[1] M-F. Horng *et al.*, ”An Adaptive Approach to Weighted Fair Queue with QoS Enhanced on IP Network”, *IEEE Catalogue No. 01CH37239*, 2001 IEEE Press.

[2] O. Gomzikov, T. Hämäläinen, and J. Joutsensalo, ”QoS Guarantee with Adaptive Scheduling Method”, *ISAST Transactions on Computers and Software Engineering*, No. 1, Vol. 1, 2007, pp.64-68.

[3] J. Joutsesnalo, T. Hämäläinen, A. Sayenko, and M. Pääkkönen, ”QoS- and Revenue Aware Adaptive Scheduling Algorithm”, *Journal of Communications and Networks*, Vol. 6, No.1 March 2004, pp. 68-77.

[4] J. Joutsensalo, T. Hämäläinen, Mikko Pääkkönen, and Alexandr Sayenko, Adaptive Weighted Fair Scheduling Method for Channel Allocation, *Proc. IEEE ICC 2003*, Anchorage, Alaska, 2003.

[5] J. Joutsensalo, T. Hämäläinen, K. Luostarinen, and J. Siltanen, “Adaptive Scheduling Method for Maximizing Revenue in Flat Pricing Scenario.”, *AEU - International Journal of Electronics and Communications*, vol. 60, issue 2, February 2006, pp. 159-167.

[6] F. P. Kelly, ”Charging and rate control for elastic traffic”. *European Transaction on Telecommunication., vol. 8*, 1997, pp. 33-37.

[7] R. J. La and V. Anantharam, ”Utility-Based Rate Control in the Internet for Elastic Traffic”. *IEEE/ACM Transactions on Networking, Volume: 10 Issue: 2*, April 2002, pp. 272–286.

[8] L. Kleinrock, *Queueing System*, John Wiley and Sons, 1975.

[9] A.K. Parekh and R.G. Gallager, ”A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case”, *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, June 1993, pp. 344-357.

[10] J. Zhang, and C. Chen, ”Implementing Scalable Service Differentiation in Cluster-based Web Server Systems”, *ISAST Transactions on Communications and Networking*, No. 1, Vol.1, 2007, pp.21-31.