Adaptive Tuning of Scheduling Parameters

Timo Hamalainen, Jarmo Siltanen, Ari Viinikainen, and Jyrki Joutsensalo
Department of Mathematical Information Technology
University of Jyvaskyla
40014 Jyvaskyla, FINLAND

ABSTRACT
This paper presents WFQ- based scheduling algorithm that allocates network resources in a fair way by taking into account class of service issues. Presented model is flexible in that different traffic flows are grouped into service classes and are given such QoS characteristics as bandwidth and delay. To adjust dynamically weights it is proposed to use the usage-based revenue criterion that enables the allocation of free resource between service classes. The analysis considers a single node and it is shown that the QoS can be guaranteed due to the allocation of unused resources to the higher service classes.

I. INTRODUCTION
To provide quality of service to traffic flows in a network, the packet scheduling discipline provided by a node (switch or router) should accommodate the various bandwidth requirements of incoming flows that share the same outgoing link. Frame scheduling becomes increasingly important when an outgoing interface is congested. To handle this situation, network administrators can assign weights to each of the different queues. This provides bandwidth to higher priority applications (e.g. using IP precedence), yet still fairly grants access to lower priority queues. The frame schedule affords each queue the bandwidth allotted to it by the network administrator.

Based on the fluid traffic model, the generalized processor sharing (GPS) discipline provides the delay and buffer occupancy bound for guaranteeing the QoS [10]. A packet-by-packet version of the algorithm, known as PGPS or Weighted Fair Queuing (WFQ), is defined in terms of the GPS system [1], [10]. A serious problem with this approach is its computational complexity. Various variants of WFQ have been developed to address this problem. On the other hand, if a fixed length packet is used, a simple round robin discipline such as the weighted round robin (WRR) could be used. Since the WRR has a larger delay bound, to alleviate this, several modification approaches of the WRR have been proposed. An another QoS based queuing method is class based queuing (CBQ). By using CBQ and link-sharing mechanisms the link bandwidth can be divided to a number of different classes, and the traffic can be isolated between these classes. If individual traffic sources within a class are to be given a service guarantee, an additional acceptance procedure must be deployed. This procedure will provide a probabilistic, and less conservative, delay bound than that provided by sorted priority algorithms [2]. To address scheduling and congestion problem more widely, [12] and [13] propose a network architecture and an algorithm, called Core-Stateless Fair Queueing (CSFQ), that reduces the implementation complexity and still achieves approximately fair allocations. Their architecture differentiates between edge and core nodes. Edge nodes handle per flow management and core nodes do not have this functionality.

Due to that differentiation the proposed model can be efficiently implemented at high speeds, and edge nodes themselves are simpler than regular fair queuing nodes.

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 a known user utility function and establish an optimization model to either maximize the user benefit or provider revenue [6], [8]. 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. An alternative is to create a market environment and let users bid for the network usage [9], [7]. This type of approach does not assume a known user utility function and achieves economic efficiency perfectly. However, the implementation overhead is quite significant in this case.

Our mechanism is designed to operate at an optimal point which maximizes the system performance in terms of the number of users served and the promised QoS requirements. 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 [3]. The possibility of using revenue as the criterion for updating weights in the WFQ service discipline for the single-node case was theoretically considered in [5], [4], and the simulation for more complex network environment was presented in [11]. The rest of the paper is organized as follows. In Section 2, used scheduling mechanism is presented and generally defined, in addition, Call Admission Control mechanism as well as some upper bounds are presented in this section. Simulations and obtained results are shown in Section 3. Final section contains discussion and future work.

II. WFQ BASED SCHEDULING ALGORITHM
Here the algorithm 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 in Fig. 1. The
data packets may have different sizes. The number of service classes is denoted by \( m \). In WFQ, the real processing time (delay) is

\[ d = N_i E(b_i) d_0 / w_i \]  

(1)

where \( w_i(t) = w_i, i = 1, \ldots, m \) are weights allotted for each class, \( N_i(t) = N_i \) is a number of customers in the \( i \)th queue, and \( E(b_i) \) is the average packet length in the queue \( i \). Here time index \( t \) has been dropped for convenience. The constraint for the weights are

\[ w_i > 0 \]  

(2)

and

\[ \sum_{i=1}^{m} w_i = 1. \]  

(3)

If some weight is \( w_j = 1 \), then the other weights are \( w_j = 0, j \neq i \), and class \( i \) is served by time \( E(b_i) d_0 \), if \( N_i = 1 \). For each service class, a revenue or pricing function

\[ r_i(d) = r_i(N_i E(b_i) d_0 / w_i + c_i) \]  

(4)

(euros/minute) is decreasing with respect to the delay \( d \). Here \( c_i(t) = c_i \) includes insertion delay, transmission delay etc., and here it is assumed to be constant. A goal is to maximize revenue criterion

\[ F(w_1, \ldots, w_m) = \sum_{i=1}^{m} N_i r_i(N_i E(b_i) d_0 / w_i + c_i) \]  

(5)

under the weight constraint 2 and 3. As a special case, consider linear revenue model.

**Definition:** The function

\[ r_i(t) = -r_i t + k_i, \quad i = 1, \ldots, m, \]  

(6)

\[ r_i > 0, \]  

(7)

\[ k_i > 0, \]  

(8)

is called linear pricing function.

**Theorem 1:** Consider the linear pricing function (6) and the corresponding revenue function

\[ F = F(w_1, \ldots, w_m, N_1, \ldots, N_m) = \sum_{i=1}^{m} N_i (-r_i E(b_i) / w_i) + k_i, \]  

(9)

where \( d_0 = 1 \) and \( c_i = 0 \) for convenience. Then upper bounds for buffer sizes are

\[ q_i = \left[ \frac{1}{2 r_i E(b_i)} \right], \quad i = 1, \ldots, m. \]  

(10)

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:

\[ \frac{\partial F}{\partial N_l} = -2 \frac{r_l E(b_l)}{w_l} N_l E(b_l) + k_l = 0. \]  

(11)

Therefore

\[ N_l = \frac{w_l k_l}{2 r_l E(b_l)}, \quad l = 1, \ldots, m. \]  

(12)

The second derivative is

\[ \frac{\partial^2 F}{\partial N_l^2} = -2 \frac{r_l E(b_l)}{w_l} < 0, \]  

(13)

because \( r_l > 0 \) and \( w_l \geq 0 \). Therefore \( F \) is strictly concave with respect to \( N_l, i = 1, \ldots, m \) having one and only one global maximum, which is satisfied by Eq. (12). Because \( w_i \leq 1, i = 1, \ldots, m \), then

\[ N_l \leq \frac{1}{2} \frac{k_l}{2 r_l E(b_l)}. \]  

(14)

for which Eq. (10) follows. This completes proof. Q.E.D.

The solution (12) is plausible and easy to interpret:

- When \( w_l \) is large, then it gives large weights to those buffers, where the number of customers is large to prevent too large delay to those buffers.
- Positive \( k_l \) increases revenue. It is simply positive constant vertical shift. Thus, larger \( k_l \) is, larger number of customers bring larger revenue.
- Negative \(-r_i\) in Eq. (9) has an opposite effect than \( k_i\). Thus, number of customers is inversely proportional to \( r_i\). The coefficient is a kind of a penalty term.
- When the average packet size \( E(b_i) \) is large, the number of packets should be small.

Upper bound for revenue is stated as follows:

**Theorem 2:** In the case of linear pricing model (6), upper bound for revenue is

\[ F \leq 4 \sum_{i=1}^{m} \frac{k_i^2}{r_i E(b_i)} \]  

(15)

**Proof:** Select optimal value for \( N_i \) in Eq. (12), and substitute it in Eq. (9). Then

\[ F = \sum_{i=1}^{m} \frac{w_i k_i}{2 r_i E(b_i)} \left( -r_i \frac{1}{2 r_i w_i E(b_i)} + k_i \right) = \frac{1}{4} \sum_{i=1}^{m} \frac{w_i k_i^2}{r_i E(b_i)} \]  

(16)
Due to the condition \( w_i \leq 1 \), Eq. (15) follows. **Q.E.D.**

Interpretation of (15) is quite obvious: \( k_i \) increases upper limit, while \( r_i \) decreases it.

As a special case, when all buffers are full according to the rule (10), we get the following result:

**Theorem 3:** When

\[
N_i = \frac{1}{2} \frac{k_i}{r_i E(b_i)},
\]

revenue is

\[
F = \frac{1}{2} \sum_{i=1}^{m} \frac{k_i^2}{r_i E(b_i)} (1 - \frac{m}{2}).
\]

Proof is omitted. It is clear that in practice the buffer sizes must be selected smaller than in Eq. (10). As a special case, when the there is only one class, i.e. \( m = 1 \), upper bound (15) can be achieved, but not for the other values of \( m \), if the buffers are full.

The following theorem gives sufficient condition for achieving non-negative revenue as well as an other upper bound for revenue:

**Theorem 4:** If weights are selected by

\[
w_i = \frac{N_i E(b_i) r_i/k_i}{\sum_{i=1}^{m} N_i E(b_i) r_i/k_i},
\]

and a constraint

\[
\sum_{i=1}^{m} \frac{N_i E(b_i) r_i}{k_i} < 1,
\]

is used in the call admission control mechanism, then

\[
0 \leq F \leq \sum_{i=1}^{m} N_i k_i.
\]

**Proof:** Define

\[
a = \sum_{i=1}^{m} N_i E(b_i) r_i/k_i.
\]

Revenue is

\[
F = \sum_{i=1}^{m} \left( -r_i N_i^2 E(b_i) \frac{k_i a}{N_i E(b_i) r_i} + N_i k_i \right)
\]

\[
= \sum_{i=1}^{m} (-N_i k_i a + N_i k_i)
\]

\[
= \sum_{i=1}^{m} N_i k_i (1 - a)
\]

\[
= \sum_{i=1}^{m} N_i k_i \left( 1 - \sum_{i=1}^{m} \frac{N_i E(b_i) r_i}{k_i} \right) \geq 0,
\]

when constraint (20) is satisfied. Because \( N_i \geq 0 \), \( r_i > 0 \), \( k_i > 0 \), then \( 0 \leq a < 1 \). Then it follows that

\[
F = \sum_{i=1}^{m} N_i k_i (1 - a) < \sum_{i=1}^{m} N_i k_i.
\]

This completes proof. **Q.E.D.**

Call Admission Control mechanism can be made by simple hypothesis testing without assumptions about call or dropping rates. Let the state (number of packets) at the moment \( t \) be \( N_i(t) \), \( i = 1, \ldots, m \). Let the new hypothetical state at the moment \( t + 1 \) be \( \tilde{N}_i(t+1) \), \( i = 1, \ldots, m \), when one or several calls appear. In hypothesis testing, hypothetical revenues \( F(t) \) and \( \tilde{F}(t) \) are calculated. If \( F(t) > \tilde{F}(t) \), then call is rejected, otherwise it is accepted.

**III. SIMULATIONS AND OBTAINED RESULTS**

In the experiment, calls and durations are Poisson and exponentially distributed, respectively. In addition, number of classes is \( m = 3 \). Data packets have lengths 1, 2, and 5 kbytes with equal probability. 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 durations are

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

The number of unit times in the experiment was \( T = 3000 \). Three service classes have the pricing functions

\[
r_1(t) = -5t + 200
\]

for gold class,

\[
r_2(t) = -2t + 100
\]

for for silver class, and

\[
r_3(t) = -0.5t + 50
\]

for bronze class.

Figures 2, 3, and 4 show the simulation results. In figure 2, three delay profiles are represented, while in Fig. 3, number of users is shown. Most importantly, figure 4 shows that the revenue is always clearly positive, justifying Theorem 4.
IV. CONCLUSIONS

In this paper, we introduced an adaptive WFQ algorithm, which adjusts dynamically weights in such a manner that the QoS requirements of the different traffic classes can be met and the network operator’s revenue can be kept as high as possible. The experiments demonstrated this property, while still allocating delays in a fair way. Here we investigated linear pricing scenario, also piecewise linear model, e.g. flat pricing model can be investigated.

In the near future, new connection admission control and queuing techniques (ie. dynamic dropping) issues will be integrated to our model. Our goal is also to study the possibility of implementation of the proposed model under the switch (eg. linux based).

REFERENCES