Gradient Scheduling Algorithm for Fair Delay Guarantee in Logarithmic Pricing Scenario

Pete Räsänen
Department of Mathematical Information Technology
University of Jyväskylä
40014 University of Jyväskylä, FINLAND
peter@cc.jyu.fi

Simo Lintunen
Department of Mathematical Information Technology
University of Jyväskylä
40014 University of Jyväskylä, FINLAND
stlintun@cc.jyu.fi

Riku Kuismanen
Department of Mathematical Information Technology
University of Jyväskylä
40014 University of Jyväskylä, FINLAND
rtkuisma@cc.jyu.fi

Jyrki Joutsensalo
Department of Mathematical Information Technology
University of Jyväskylä
40014 University of Jyväskylä, FINLAND
jyrkij@cc.jyu.fi

Timo Hämäläinen
Department of Mathematical Information Technology
University of Jyväskylä
40014 University of Jyväskylä, FINLAND
timoh@cc.jyu.fi

ABSTRACT

In this paper we propose a packet scheduling scheme for ensuring delay as a Quality of Service (QoS) requirement. For customers, fair service is given while optimizing revenue of the network service provider. Gradient type algorithm for updating the weights of a packet scheduler is derived from a revenue-based optimization problem in the logarithmic pricing scenario. Algorithm is simple to implement. We compared algorithm with optimal brute-force method. The weight updating procedure is independent on the assumption of the connection's statistical behavior, and therefore it is robust against erroneous estimates of statistics.

Keywords

Pricing, revenue optimization, delay, bandwidth, Quality of Service (QoS)

1. 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.

QOSIM 2008, March 03, Marseille, France

DOI 10.4108/ICST.SIMUTOOLS2008.3046

Based on the fluid traffic model, the generalized processor sharing (GPS) discipline provides the delay and buffer occupancy bound for guaranteeing the QoS [1]. 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, 2]. 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. 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 [3]. To address scheduling and congestion problem more widely, [4] and [5] 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 the approaches assume a known user utility function and establish an optimization model to either maximize the user benefit or provider revenue [6, 7]. 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 difficully 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 [8, 9]. This type of approach does not assume a known user utility function

Figure 1: An example of a packet scheduler with two classes

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. 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 [10, 11]. In these cases, simple closed form updating rules were developed. However, in this paper we consider logarithmic pricing, which leads to quite simple gradient type updating rule for optimizing revenue.

2. THE PACKET SCHEDULER AND DELAYS

In this section, we formulate expressions for delays of the data traffic. Consider the packet scheduler in Fig. (1). There are now two service classes. The Gold class customers pay most of the money while getting best service, and the Silver class customers pay least of the money.

Parameter Δt_i denotes time which passes when data is transferred through the queue i to the output in the switch, when $w_i = 1$. If the queue is almost empty, delay is small, and when the buffer is full, it is large. Variable w_i is the weight allocated for class i. Constraint for weights w_i is

$$\sum_{i=1}^{m} w_i = 1, \quad w_i > 0. \tag{1}$$

Variables w_i give weights, how long time queues i are served per total time. Therefore, delay d_i in the queue i is actually

$$d_i = \frac{\Delta t_i}{w_i}. \tag{2}$$

Without loss of generality, only non-empty queues are considered, and therefore

$$w_i \neq 0, \quad i = 1, \ldots, m, \tag{3}$$

where m is number of service classes. When one queue becomes empty, $m \to m - 1$. Parameter N_i denotes the number of such connections in the ith service class.

3. PRICING MODELS AND REVENUE MAXIMIZATION

We concentrate on the pricing and fair resource allocation from the point of view of the customers. On the other hand, from the point of view of the service provider, we try to maximize revenue. First, we introduce the concept of *pricing functions*.

For delay, pricing functions are denoted by $f_i(d)$, where d is the delay, and f_i is decreasing with respect to d. In addition, $f_i(d)$ is (strictly) convex with respect to d for ensuring revenue more than $-\infty$. Revenue obtained for one user in the class i is just $f_i\left(\frac{\Delta t_i}{w_i}\right)$. Because there are N_i connections in the class i with all having the same delay, revenue corresponding to the delays in the class i is

$$R_i^{delay}(w_i) = N_i f_i\left(\frac{\Delta t_i}{w_i}\right). \tag{4}$$

In our study, we use *logarithmic* pricing function for delays. Then pricing function has the form

$$f_i(d) = -\log(a_i d + b_i), \quad a_i > 0, \quad 0 < b_i < 1. \tag{5}$$

When delay is $d = 0$, f has the form

$$f_i(0) = -\log(b_i). \tag{6}$$

Because $0 < b_i < 1$:

$$f_i(0) > 0. \tag{7}$$

But this is desirable feature. If we denote higher priority classes by smaller index, we have

$$b_i < b_j, \quad i < j. \tag{8}$$

$$a_i > a_j, \quad i < j. \tag{9}$$

Revenue is

$$R = -\sum_{i=1}^{m} N_i \log_c\left(\frac{a_i \Delta t_i}{w_i} + b_i\right) + \lambda(1 - \sum_{i=1}^{m} w_i), \tag{10}$$

where λ is a Lagrangian penalty term. The gradient has the form

$$\frac{\partial R}{\partial w_i} = \frac{N_i a_i \Delta t_i}{\ln c(a_i \Delta t_i w_i + b_i w_i^2)} - \lambda. \tag{11}$$

Due to the constraint for the weights

$$\lambda = \lambda \sum_{k=1}^{m} w_k = \sum_{k=1}^{m} \lambda w_k. \tag{12}$$

Then we get

$$\frac{\partial R}{\partial w_i} = \frac{N_i a_i \Delta t_i}{a_i \Delta t_i w_i + b_i w_i^2} - \sum_{k=1}^{m} \frac{N_k a_k \Delta t_k}{a_k \Delta t_k + b_k w_k}. \tag{13}$$

Concavity of the function is seen from the second order derivative. It is

$$\frac{\partial^2 R}{\partial w_i^2} = \frac{N_i a_i \Delta t_i}{(a_i \Delta t_i + b_i w_i)^2}\left(b_i - \frac{2b_i}{w_i} - \frac{a_i \Delta t_i}{w_i^2}\right) < 0. \tag{14}$$

So R is concave, having strictly one maximum.

Because gradient shows the direction of largest increase of the function R with respect to the variables w_i, the weights are updated according to the gradient algorithm. During one iteration step, weights are updated according to the rule

1. Update weights

$$v_i(t) = w_i(t) + \mu \left.\frac{\partial R(N_1(t), \ldots, N_m(t), w_i)}{\partial w_i}\right|_{w_i = w_i(t)}. \tag{15}$$

2. Perform scaling

$$w_i(t+1) = \frac{v_i(t)}{\sum_{j=1}^{m} v_i(t)}. \tag{16}$$

Here t denotes time, and $N_i(t)$ shows that the number of connections vary as a function of time. Parameter μ is a forgetting factor, which is either constant or depend inversely on the norm of the gradient. It ensures time-varying nature of the weights due to the nonstationary data traffic.

4. EXPERIMENTS

4.1 Setup

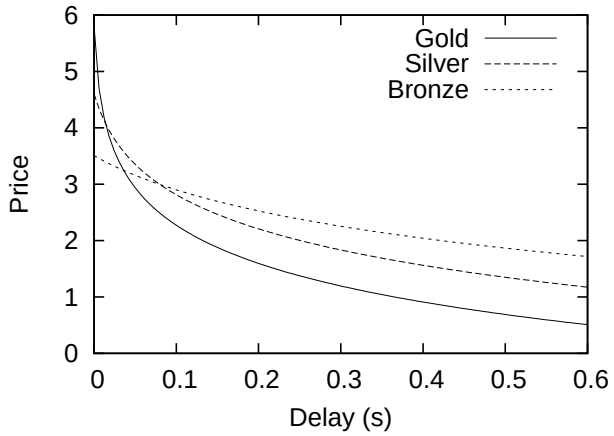


Figure 2: Pricing functions for each class

The simulations were carried out on the NS-2 network simulator [12] using the implementation of the Adaptive WFQ -scheduler implemented in [13]. AWFQ-scheduler was used to update weights dynamically every 0.1 seconds.

We compared gradient based approach to optimal brute force method, in which weights were chosen by calculating all possible weight combinations with a grid resolution of 0.01 and choosing the one that gives the highest revenue. In the gradient based approach, iteration step defined in formulas (15) and (16) was used to calculate new weights. Ten iterations were performed.

All client applications are divided into three different pricing classes: Gold, Silver and Bronze respectively. Figure 2 shows logarithmic pricing curves used for each class. We can see from the figure that the Gold class customers pay the most money when the delay is minimal, but the price also quickly reduces as the delay increases. For the Silver and the Bronze classes the price is lower with low delay but the price does not reduce as fast as in the Gold class when the delay increases. When delay increases the Bronze class becomes the most expensive one.

The topology of the simulated network is shown in Figure 3. It consists of a router with the Adaptive WFQ -scheduler, a destination server and a set of client nodes. The topology was kept simple since the idea is to simulate how the implemented packet scheduler behaves when there is one so called bottle neck link. In this topology it is simpler to manage conditions so that bottle neck link is shared between traffic of all classes based on algorithm's results. This way we can analyze the algorithm's performance in controllable situations. It should be noted that the simulation scenario does not correspond to real-life conditions when the traffic medium is not always this congested. The main purpose of this research is to study the behaviour of the algorithm in very congested conditions.

Details for each service class are shown in Table 1. All traffic is transferred using the UDP protocol. A and b are parameters for

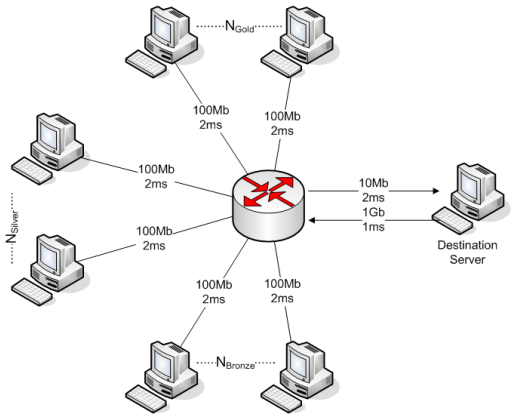


Figure 3: The topology of the simulated network.

logarithmic pricing function (5). Each connection was set to start and stop sending data periodically.

Table 1: The characterics of the traffic

	Gold	Silver	Bronze
a	1	0.5	0.25
b	0.03	0.1	0.3
Num of Conn.	0-40	0-30	0-20
Queue length	50	75	100
Type	UDP		
Bandwidth	256 kB		
Packet size	1000 B		

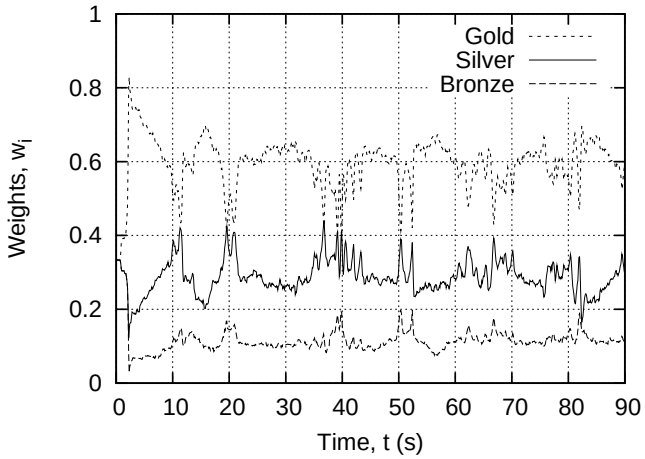
4.2 Results

Simulation results are presented in Figure 4. In the very start of the brute force methods results, we can see that it reacts much faster to the rapid rise of the amount of the customers. On the other hand it causes the peak delay of lower classes to rise above 1800 milliseconds (see Figure 4f) when the gradient approach keeps the delays below 1400 milliseconds (see Figure 4e).

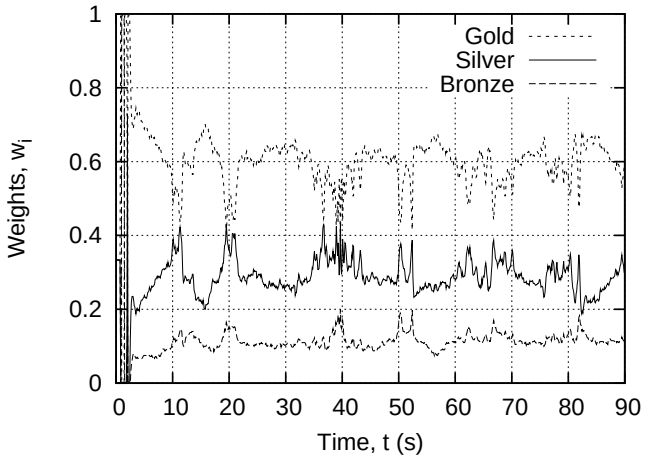
In figures 4a and 4b are presented the evolution of weights with given parameters in both gradient and brute force approach. Weight update intervall was 0.1 seconds. As we can see, there are only minor differences between the weights at each time.

Bandwidth in figures 4c and 4d show that our gradient approach follows the optimal solution with minor differences. Bandwidth for the Silver and Bronze class rises at approximately 40 and 80 seconds. This can be explained by noticing that number of flows is low at these times (see Figure 5). At 40 seconds delay also drops substantially for each class. For the Gold class delay stays always on a good level considering e.g. a real-time video conversation or a VoIP-call.

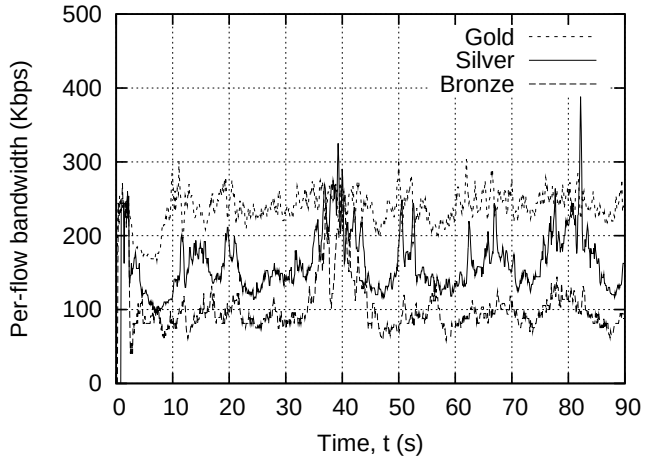
Figure 6 shows the evolution of the revenue. The final revenue yielded by the algorithm is $R = 218908$. On the other hand, brute-force method using 1/1000 resolution grid ($w_i = 0.001, ..., 0, 998$) is $R = 221547$. Thus $R_{adaptive}/R_{bruteforce} = 99\ \%$ in ten iterations. Thus we can conclude that our adaptive mechanism produces quite optimal revenue by using simple computations - brute-force method has exponential complexity.



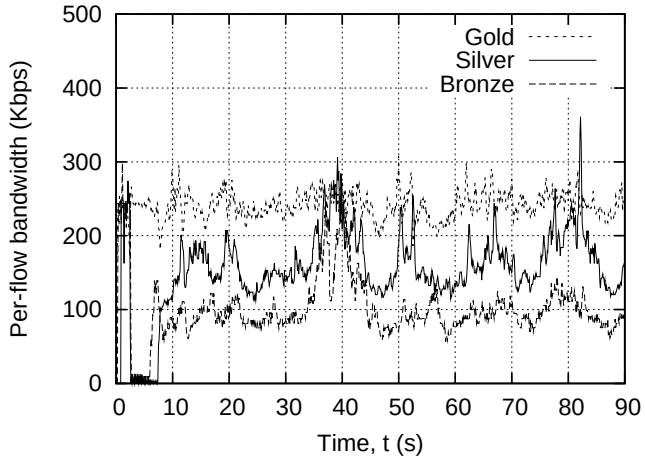
(a) The evolution of weights (Gradient)



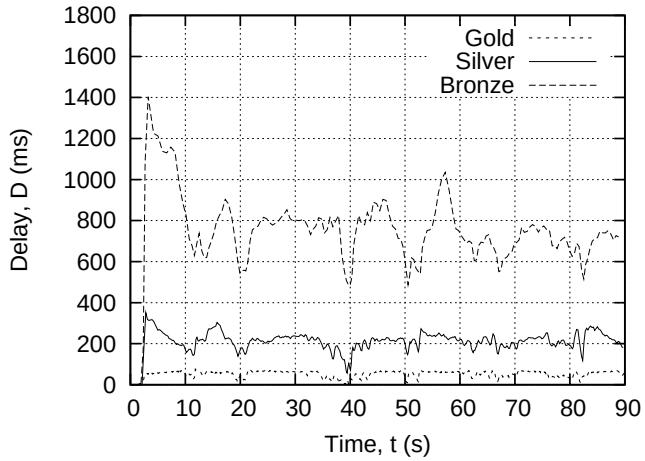
(b) The evolution of weights (Brute)



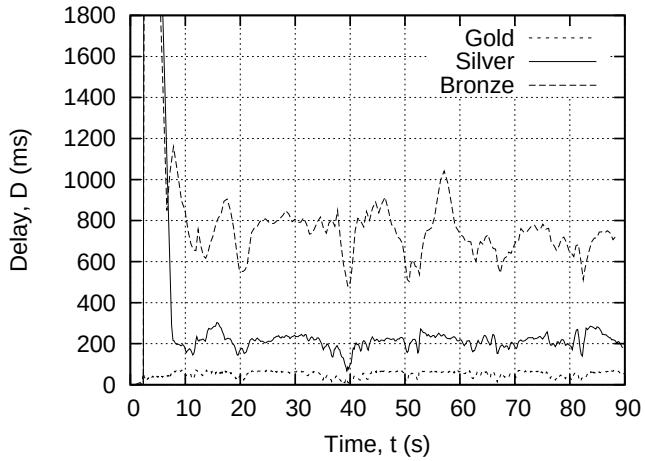
(c) Per-flow bandwidth of each class (Gradient)



(d) Per-flow bandwidth of each class (Brute)



(e) Delay of each class (Gradient)



(f) Delay of each class (Brute)

Figure 4: Simulation results.

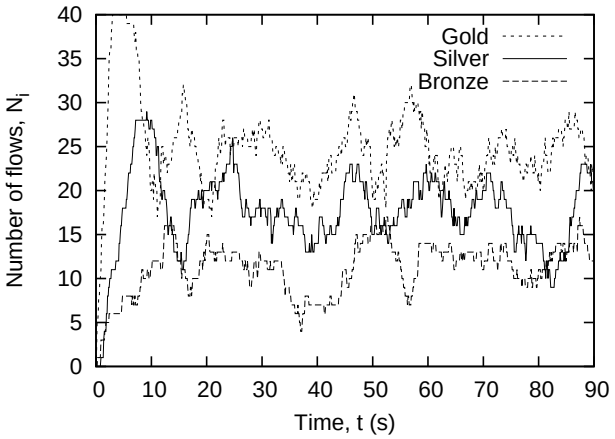


Figure 5: Evolution of flows.

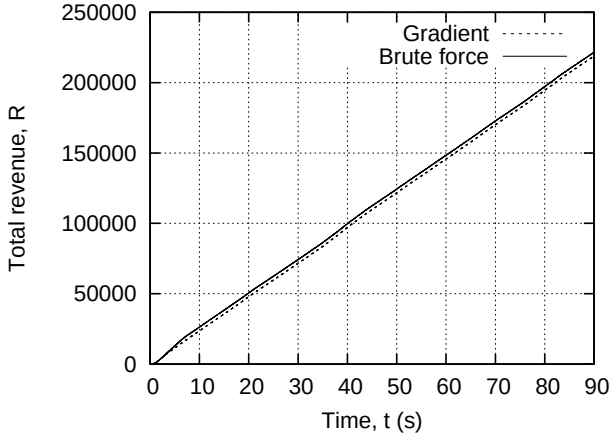


Figure 6: Revenue curve of the gradient and brute force approach as a function of time.

5. DISCUSSION

We make the following conclusions shown by algorithm and experiments:

- In the pricing scenario, we have derived analytic form to the revenue and gradient algorithm for updating the weights w_i, which allocate data traffic to the connections of different service classes.
- The updating procedure is deterministic, 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 the wrong models.
- Algorithm is unique and optimal, which have been proved by Lagrangian optimization method.
- Minimum bandwidth can be guaranteed in call admission mechanism by adding a specific constraint to the updating rule. This feature is studied later.
- Because gain factors are positive, all classes obtain service in fair way.

6. CONCLUSIONS

In this paper, we presented adaptive algorithm for optimizing the network operator revenue and for ensuring delay as QoS requirement. We derived the algorithm from a revenue-based optimization problem. Also CAC mechanism can be used in the connection level. In the experiments we simulated the operation of the adaptive algorithm and compared it with a brute-force method. The obtained results show that by using the adaptive weights the revenue is the same than with the brute-force method. Our algorithm is deterministic, and thus we believe that in practical environments it is a competitive candidate due to its robustness.

7. REFERENCES

[1] 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.

[2] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a fair queuing algorithm," IEEE SIGCOM'89, pp. 1–12, 1989.

[3] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks", *IEEE/ACM Transactions on Networking*, Volume: 3 Issue: 4 , Aug. 1995, Page(s): 365–386.

[4] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queuing: Achieving approximately fair bandwidth allocations in high speed networks". *Proc. SIGCOMM 1998*, Canada, Sept. 1998.

[5] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management". *Proc. SIGCOMM 1999*, Boston, Sept. 1999.

[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] S. H. Low, "Equilibrium bandwidth and buffer allocations for elastic traffics". *IEEE/ACM Transactions Networking, vol. 8*, June 2000, pp. 373–383.

[9] Y. A. Korilis, A. A. Lazar, and A. Orda, "Achieving network optima using Stackelberg routing strategies". *IEEE/ACM Transactions on Networking, vol. 5*, Feb. 1997, pp. 161âĂŞ-173.

[10] Jyrki Joutsensalo, Timo Hämäläinen, Alexander Sayenko, and Mikko Pääkkönen, "QoS- and Revenue Aware Adaptive Scheduling Algorithm", *Journal of Communications and Networks*, Vol. 6, No.1 March 2004, pp. 68–77.

[11] Jyrki Joutsensalo, Timo Hämäläinen, Kari Luostarinen, and Jarmo 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.

[12] UCB/LBNL/VINT. Network simulator ns-2. http://www.isi.edu/nsnam/ns.

[13] A. Sayenko, "Adaptive scheduling for the QoS supported networks", Dissertation, University of Jyväskylä, 2005, 216 p.