Paweł Świątek
Institute of Computer Science
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland
Email: pawel.swiatek@pwr.wroc.pl

Adam Grzech
Institute of Computer Science
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland
Email: adam.grzech@pwr.wroc.pl

Piotr Rygielski
Institute of Computer Science
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland
Email: piotr.rygielski@pwr.wroc.pl

Adaptive packet scheduling for requests delay guaranties in packet-switched computer communication network

Abstract. In this paper the problem of packet scheduling in the node of packet-switched computer communication network is considered. Packet scheduling in the network edge nodes is one of the crucial mechanisms essential for delivery of required level of quality of end-to-end network services (*QoS*). In order to satisfy *QoS* guaranties for each incoming request belonging to one of distinguished traffic classes, packet scheduling algorithm must make decisions based on current state of the scheduling system (e.g. buffer lengths) and actual characteristics of the serviced traffic (e.g.: lengths of connections following requests, packet intensities within requests, etc.).
The paper is devoted to propose a new packet scheduling algorithm based on *Weighted Round Robin* (*WRR*), which weights are adapted according to changes of network's load and traffic characteristics. By means of computer simulation, on representative examples, it was shown, that utilization of additional knowledge about incoming traffic in the process of packet scheduling may improve *QoS* guaranties for serviced traffic.

Keywords. QoS, packet scheduling, adaptation.

1. Introduction

One of the most important mechanisms for delivering the quality of end-to-end network services (*QoS*) in packet-switched computer communication networks is packet classification and scheduling in network nodes. Delivering *QoS* consists of guaranteeing for each separate stream of packets (e.g. connection) certain values of end-to-end network services'parameters, such as: maximum or average packet delay, jitter, packet loss ratio, etc. Required values of network services parameters depend on the traffic class, to which separate streams of packets belong to, and on amount of resources devoted to service the distinguished traffic class. Traffic classes are often distinguished basing on applications, which generate the traffic. *QoS* requirements are, in general, different for different traffic classes and depend on specific values of network services parameters required by various applications, necessary for them to run correctly. The task of packet scheduling algorithm is to assure required service for packets belonging to different streams in such a way, that *QoS* requirements are met for each separate stream of packet [6,12]. In further parts of the paper it is assumed that the incoming traffic is a result of accepted requets and takes form of connections composed of randomly distributed sequences of packets [6]

In order to satisfy *QoS* requirements of each separate stream of packets, scheduling decisions must be made basing on current state of the scheduling system (e.g. buffer lengths, requests and packets incoming rate) and actual characteristics of the serviced traffic (e.g.: lengths of connections generated by separate requests, packets intensities within connections, etc.) [3]. Unfortunately, most of incoming traffic characteristics are not available to the scheduling algorithm at the moment when proper decision is required. Moreover, complexity of the scheduling algorithm increases with the number of parameters taken into account in the decision making process, what strongly affects efficiency of the scheduling algorithm and systems throughput [5,11].

To overcome these difficulties, we propose a new scheduling algorithm, which bases on estimation of traffic characteristics and adaptation of *Weighted Round Robin* (*WRR*) algorithm. The main advantage of such an approach relies on facts, that scheduling and adaptation are two separate processes and that single adaptation step may be several orders of magnitude longer, than step of primary scheduling algorithm. In fact, the duration of single adaptation step is bounded by the frequency of major changes of the traffic characteristics. Therefore, utilization of simple *WRR* as the primary scheduling algorithm enables fast decision making and high systems throughput, while more complex adaptation provides required level of *QoS*. Additionally, estimation of traffic characteristics allows to improve the quality of scheduling by prediction of future values of traffic parameters (such as connection lengths)[8].

2. Traffic model

The discussed scheduling tasks are closely related to incoming packets processing in networks nodes.

Traffic generated at sources and composed of sequences of multipacket requests are multiplexed and transmitted over transmission trunks to the edge network nodes (Figure 1).

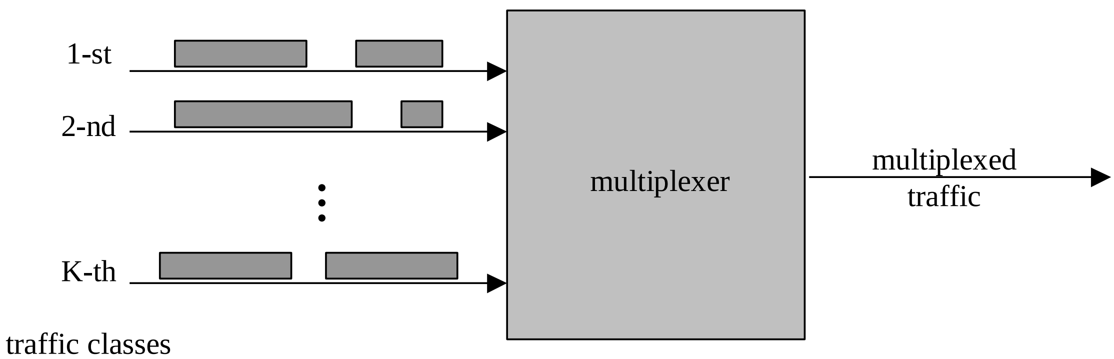

Figure 1. Incoming traffic processed in the network edge node

At the network domain edge node the incoming traffic is processes at classifier (selects a packet from a traffic stream based on the content of some portion of the packet header), meter (checks compliance to traffic parameters (i.e.: token bucket)) and passes results to the marker and shaper/dropper to trigger action for in/out-of-profile packets (delays some packets to be compliant with the previously specified traffic profile) (Figure 2).

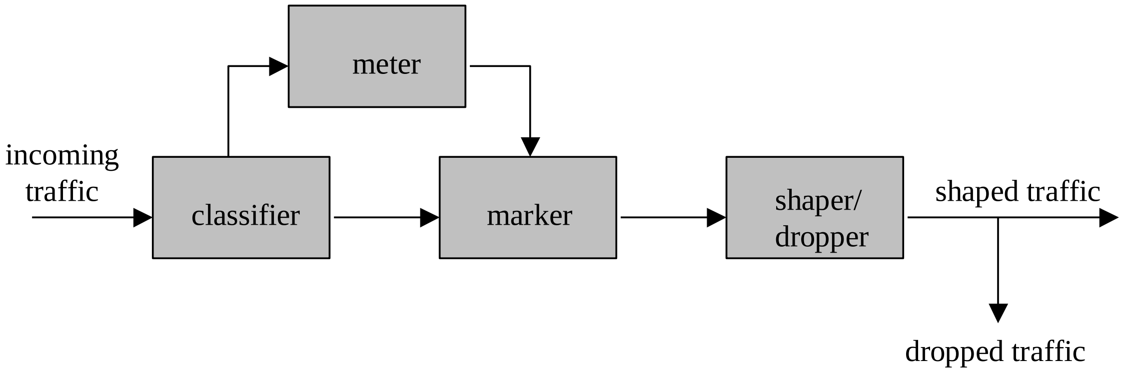

Figure 2. Incoming traffic processed in the network edge node

The shaped traffic (in form of sequences of packets belonging to connections uniquely identified by requests) is then allocated in queues (Figure 3).

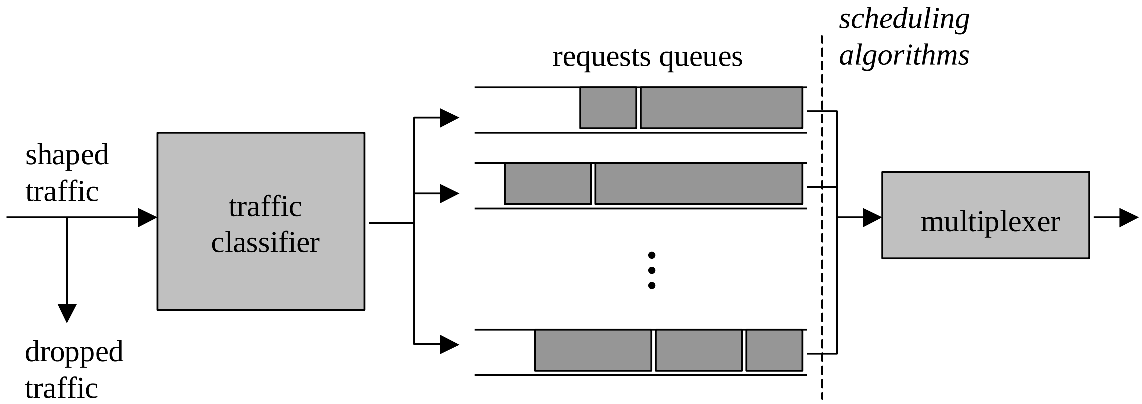

Figure 3. Classification and scheduling of requests

It is assumed that the classifier recognizes connections and allocates them as a whole in one of available queues representing distinguished and supported traffic classes. The general idea of the proposed attempt is based on assumption that the quality of services offered for requests depends on proper recognition of the lengths of connections (sequences of packets following requests). The discussed requests, connections and packets service processing assures original sequence of packets, i.e., all packets are processed in all nodes at the route from source to destination in order in which the packets were generated by source.

For the purpose of this paper, it is assumed, that the aggregated stream of packets incoming into the network boundary (edge) node is composed of multiplexed substreams of packets belonging to requests uniquely defined by the source and destination addresses as well as class of traffic to which it belongs. The multiplexed requests may be considered both based on source – destination addresses as well as number of traffic classes. For purpose of further analysis it is assumed that the different source – destination addresses are equivalent to different traffic classes.

Each source of traffic generates requests belonging to one of certain number (say K) of distinguished traffic classes. Each j-th request from $k-th$ class (denoted by req_{kj}) : class number $k \in \{1,...,K\}$, arrival time t_{kj}, duration τ_{kj} and the distribution of packets within the request expressed also by sequence of arrival times of packets belonging to this request $\{t_{kj1},...,t_{kji_{kj}}\}$ (Figure 4).

$$req_{kj} = <k, t_{kj}, \tau_{kj}, \{t_{kj1},...,t_{kji_{kj}}\}>, \tag{1}$$

where i_{kj} is the number of packets within request req_{kj}.

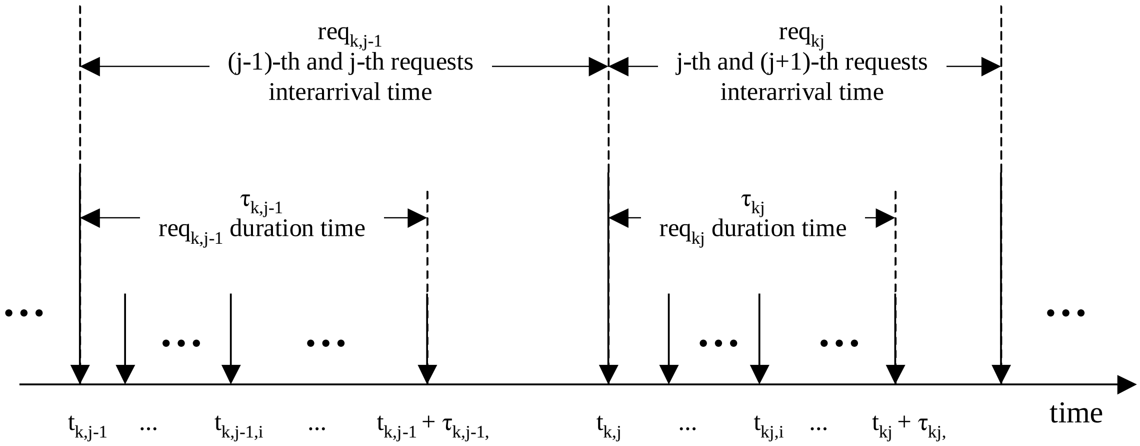

Figure 4. Requests from k-th traffic class

Sequences arrival times of successive requests ($\{t_{k1},...,t_{kj},...\}$), as well as sequences of arrival times of successive packets within these requests ($\{t_{kj1},...,t_{kji_{kj}}\}$) are specific to requests belonging to the same k-th traffic class. The behavior of the source generating traffic belonging to particular traffic class may be described by certain more or less complex probability distribution. In the simples case it can be an exponential distribution characterized by single parameter. In other cases, however, traffic may be generated by *ON*/*OFF* sources described by modulated stochastic processes or it can be self-similar traffic. The latter class of traffic is characteristic for digital telephony and interactive apllications

Each incoming request req_{kj} belongs to one of the K distinguished traffic classes ($k \in \{1,...,K\}$). The number K of traffic classes depends on the way classes are defined. K may for example be equal to: the number of request currently being serviced in the node, the number of classes distinguished in the quality of service policy assumed in the network, or to the number of traffic processing methods related to various tasks network security policy enforcement.

Generally, in the aggregated stream of packets incoming to the network node substreams of packets (requests) belonging to different traffic classes can be distinguished. Traffic classification is generally related to different ways of processing requests from different traffic classes and/or different quality of services requirements of various traffic classes.

Traffic classification has most often a hierarchical nature. In a typical scenario one can distinguish following hierarchical levels: overall traffic generated by single organization, traffic generated by certain type of application and by certain user. At the lowest level of granularity a single request generated by a single usage of particular application can be treated as separate traffic class. The behavior of such a traffic class can be characterized by three probability distributions functions $f_{k\delta}(\boldsymbol{\delta}_k)$, $f_{k\tau}(\boldsymbol{\tau}_k)$ and $f_{k\alpha}(\boldsymbol{\alpha}_k)$ describing respectively: time interval δ_{kj} between arrival of two consecutive requests (req_{kj-1} and req_{kj}), duration τ_{kj} of request and time interval α_{kji} between arrival of two consecutive packets within single request. Vectors $\boldsymbol{\delta}_k$, $\boldsymbol{\tau}_k$ and $\boldsymbol{\alpha}_k$ are parameters of corresponding distribution functions and are specific to particular k-th traffic class. On the higher level of granularity traffic classes can be modeled similarly or as the composition (aggregation) of traffic generated by classes at lower levels.

Sequence of requests from k-th traffic class is presented at the Figure 4. According to the traffic model introduced above sequences: $\{t_{kj}\}, \{\tau_{kj}\}$ and $\{\{t_{kj1},...,t_{kji_{kj}}\}\}$ related to requests from k-th traffic class can be described by distribution functions: $f_{k\delta}(\boldsymbol{\delta}_k)$, $f_{k\tau}(\boldsymbol{\tau}_k)$ and $f_{k\alpha}(\boldsymbol{\alpha}_k)$. Therefore parameters characterizing requests from the same k-th traffic class are realizations of random variables described by the same probability distributions.

3. Adaptive packet scheduling

3.1. Problem formulation

Assume following model of the network node as the multi-queue single-processor queuing system (fig. 5). Network node consists of processing unit ***P*** and K queues q_k, each buffering connections from corresponding traffic classes.

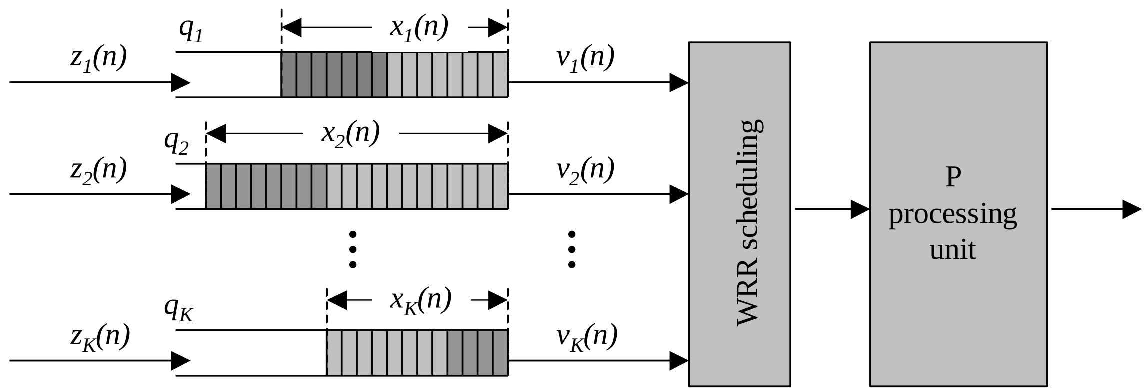

Figure 5. Network node as the multi-queue single-processor queuing system

Packets from queues q_k are scheduled according to *WRR* algorithm. In each *WRR* cycle $n = 1,2,...$ processing unit ***P*** serves $v_k(n)$ packets from k-th queue. Vector $\mathbf{v}(n)$ of number of packets served from all queues is proportional to the vector $\mathbf{w}$ of *WRR* weights:

$$\mathbf{v}(n) = \mu\mathbf{w}, \tag{2}$$

where μ is the speed of processing unit ***P***.

In the same time $\mathbf{z}(n)$ packets arrive to all queues. Denote by $\mathbf{x}(n)$ and $\mathbf{x}(n+1)$ vectors of numbers of packets from each class, buffered in queues at the beginning and at the end of n-th cycle, respectively. Obviously, the state of the processors queues can be described by following flow equation [1]:

$$\mathbf{x}(n+1) = \mathbf{x}(n) - \mathbf{v}(n) + \mathbf{z}(n) = \mathbf{x}(n) - \mu\mathbf{w} + \mathbf{z}(n). \tag{3}$$

Let $\mathbf{q}(n)$ be the vector of measured delays of connections from all traffic classes in the n-th cycle. Vector $\mathbf{q}(n)$ can be calculated as certain function $\overline{h}$ of queues state $\mathbf{x}(n)$ and numbers of incoming ($\mathbf{z}(n)$) and outgoing ($\mathbf{v}(n)$) packets [10]:

$$\mathbf{q}(n) = \overline{h}(\mathbf{x}(n), \mathbf{v}(n), \mathbf{z}(n)) = h(\mathbf{x}(n), \mathbf{w}, \mathbf{z}(n)) . \tag{4}$$

The task of scheduling algorithm is to minimize certain quality of service index $\varphi(\mathbf{q}(n))$ (e.g. average traffic delay) and to guarantee required values of delay for each connection belonging to each traffic class. Denote by $\mathbf{Q}$ the vector of delay requirements of all traffic classes. Now, the task of scheduling can be formulated as the following optimization problem:

For given:

- queues state $\mathbf{x}(n)$ at the beginning of n-th cycle,
- numbers of packets $\mathbf{z}(n)$ from all classes which arrive to the system during n-th cycle,
- function $\overline{h}$ describing the influence of queue lengths $\mathbf{x}(n)$ and numbers of packets entering $\mathbf{z}(n)$ and leaving $\mathbf{v}(n)$ the system on delays of serviced connections,
- quality of service index φ .

Find:

Such a vector $\mathbf{v}^*(n)$ of scheduling decisions, which minimizes the quality of service index $\varphi(\mathbf{q}(n))$:

$$\mathbf{v}^*(n) = \arg\min_{\mathbf{v}(n)} \varphi(\mathbf{q}(n)) = \arg\min_{\mathbf{v}(n)} \varphi(\overline{h}(\mathbf{x}(n), \mathbf{v}(n), \mathbf{z}(n))) \tag{5}$$

with respect to *QoS* constraints:

$$\mathbf{q}(n) \leq \mathbf{Q} . \tag{6}$$

Since decisions $\mathbf{v}(n)$ can be affected only by changing weights $\mathbf{w}$, then, according to (4), equation (5) can be rewritten as:

$$\mathbf{w}^*(n) = \arg\min_{\mathbf{w}(n)} \varphi(\mathbf{q}(n)) = \arg\min_{\mathbf{w}(n)} \varphi(h(\mathbf{x}(n), \mathbf{w}(n), \mathbf{z}(n))) \tag{7}$$

3.2. Algorithm

The task of adaptive packet scheduling is defined by formula (7) and constraints (6). Unfortunately, it cannot be directly solved, due to the fact, that function h defining connection delays and vector $\mathbf{z}(n)$ of numbers of packets incoming to the system during the n-th cycle are unknown.

In order to approximate the solution, the problem must be decomposed into four simpler subproblems, which are iteratively solved:

1. Estimation of parameters $\boldsymbol{\delta}_k$, $\boldsymbol{\tau}_k$ and $\boldsymbol{\alpha}_k$ of probability distribution functions $f_{k\delta}(\boldsymbol{\delta}_k)$, $f_{k\tau}(\boldsymbol{\tau}_k)$ and $f_{k\alpha}(\boldsymbol{\alpha}_k)$ characterizing connections from each traffic class $k = 1,...,K$.
2. Prediction of the vector $\overline{\mathbf{z}}(n)$ of numbers of packets from all traffic classes, which arrive to the system during the n-th cycle.

3. Approximation of function h by assumed model $\Phi(\boldsymbol{\theta})$.
4. Minimization of the quality of service index $\varphi(\Phi(\mathbf{x}(n), \mathbf{w}(n), \bar{\mathbf{z}}(n); \boldsymbol{\theta}(n)))$ with respect to *QoS* constraints (6).

The above decomposition allows to utilize solution algorithms known from such fields as: estimation, prediction, identification and optimization. In this work, following algorithms were applied. In the case, when classes of distributions were known, the *maximum likelihood method* was used to estimate distribution parameters.

Analysis of stochastic processes build upon distributions $f_{k\delta}(\boldsymbol{\delta}_k)$, $f_{k\tau}(\boldsymbol{\tau}_k)$ and $f_{k\alpha}(\boldsymbol{\alpha}_k)$ allows to predict values of each element of vector $\bar{\mathbf{z}}(n)$ as the product of expected values of processes describing the number of active connections from each traffic class and the number of packets from a single connection belonging to each traffic class, which arrive to the system during one cycle. For example, when distributions $f_{k\delta}$, $f_{k\tau}$ and $f_{k\alpha}$ are exponential with parameters δ_k, τ_k and α_k, the expected value of k-th element of $\bar{\mathbf{z}}(n)$ can be calculated as [4]:

$$\bar{z}_k(n) = \Delta\alpha_k\left(\frac{\tau_k}{\delta_k}\left(1 - e^{-\frac{\Delta t}{\tau_k}}\right) + l_k(n-1)\, e^{-\frac{\Delta t}{\tau_k}}\right) \tag{8}$$

where Δt is the length of scheduling cycle and $l_k(n-1)$ is the measured number of active connections in (n-1)-th cycle. If distributions $f_{k\delta}$, $f_{k\tau}$ and $f_{k\alpha}$ are unknown, it is convenient to use adaptive *autoregressive moving average* filter (*ARMA*) as the predictor of $\bar{\mathbf{z}}(n)$.

As the model of delays of connections from all traffic classes $\bar{\mathbf{q}}(n) = \Phi(\mathbf{x}(n), \mathbf{w}(n), \bar{\mathbf{z}}(n); \boldsymbol{\theta}(n))$ a *diagonal recurrent neural network* (*DRNN*) [9] was used. Models parameters $\boldsymbol{\theta}(n)$ were approximated according to *backpropagation through time* (*BPTT*) algorithm [2].

In order to find optimal values of *WRR* weights $\mathbf{w}^*(n)$, which minimize the quality of service index $\varphi(\Phi(\mathbf{x}(n), \mathbf{w}(n), \bar{\mathbf{z}}(n); \boldsymbol{\theta}(n)))$ with respect to *QoS* constraints (6), *simulated annealing* metaheuristic was applied. However, for the cases, when the number K of distinguished traffic classes is not high ($K \le 5$), exhaustive search may be used.

The abovementioned algorithms and models applied in four step iterative process, to which the task of adaptive scheduling (7) was decomposed to, constitute the *Adaptive Weighted Round Robin*} (*AWRR*) algorithm.

4. Simulation study

In order to evaluate the quality of service delivered by proposed *AWRR* algorithm, it was compared to other known scheduling algorithms: *WRR* and *PRIO*. *WRR* is the simple weighted round robin with weights proportional to classes priorities. *PRIO* is the priority scheduling algorithm, which allocates all system resources to class with the highest priority.

In the simulations, it was assumed, that there is $K = 3$ traffic classes. Connection from each class were generated by certain number of *ON/OFF* sources. Priorities of classes were set to $p_1 = 1$, $p_2 = 5$ and $p_3 = 10$. *QoS* requirements of classes were $Q_1 = \infty$, $Q_2 = 100$ and $Q_3 = 50$. Above assumptions mean, that the first traffic class was the *best effort* traffic.

Exemplary results obtained for the *WRR* algorithm are presented on Figure 6. The chart presents average delay of connections from each traffic class during the simulation period. It is easy to notice, that proposed *AWRR* algorithm allocates to the high priority traffic class (class 3) only such amount of resources, which is necessary to deliver required level of connection delays. Remaining resources are allocated to the lower priority classes allowing them to experience lower delays.

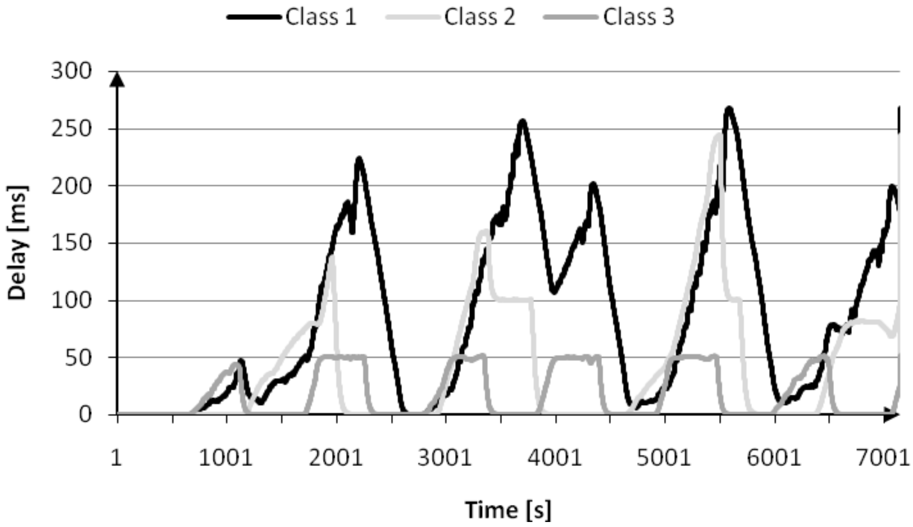

Figure 6. Connection delay for three traffic classes delivered by *AWRR*.

Unfortunately, *QoS* requirements cannot be met for all traffic classes (e.g. class 2). The reason is, that overall traffic volume incoming to the network node is sometimes higher, than nodes processing capabilities. Thus, strict *QoS* guaranties for all traffic classes cannot be delivered without additional *QoS* mechanisms [5,7], such as: admission control, traffic shaping, etc.

Each one of compared algorithms guarantied the required level of connection delays for the high priority traffic class (class 3). Moreover, for each of algorithms, requirements of second traffic class were violated, when traffic intensity was high. Therefore, the quality of scheduling of evaluated algorithms can be measured as the average delay of best effort traffic (class 1) under condition, that requirements of high priority traffic (class 3) are met.

Average delay of connections from first traffic class, delivered by compared algorithms, are presented on Figure 7.

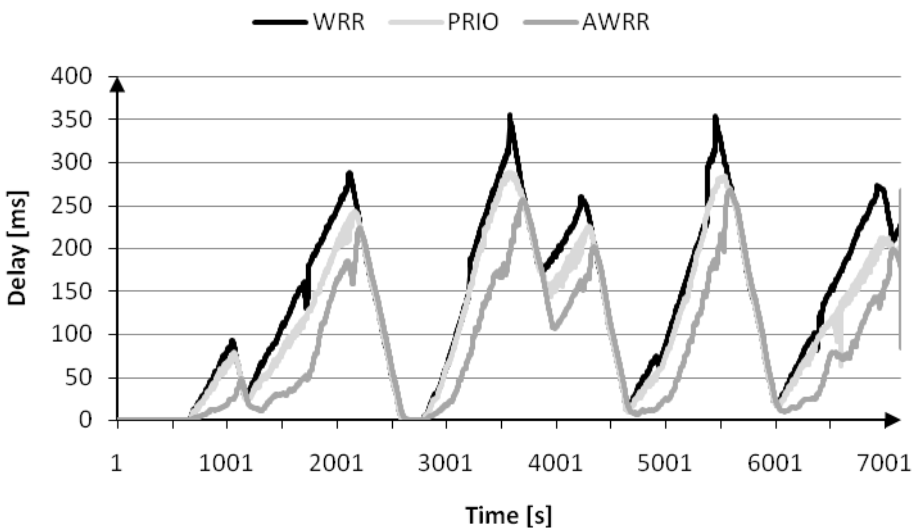

Figure 7. First class delay delivered by compared algorithms: *WRR*, *PRIO*, *AWRR*.

One can notice, that the lowest delays are obtained for the proposed *AWRR* algorithm and the highest delays are obtained for classic *WRR*. Sample results of algorithms comparison are gathered in Table 1.

Table 1. Average connection delay for three traffic classes delivered by compared algorithms: *WRR*, *PRIO*, *AWRR*.

Algorithm	Class 1 avg. delay	Class 2 avg. delay	Class 3 avg. delay
WRR	156,43	33,30	3,98
PRIO	151,10	26,98	0,03
AWRR	98,81	97,62	44,84

Both *WRR* and *PRIO* allocate systems resources to second and third traffic class and deliver low average delay for these two classes. Obviously, resources can be allocated more fairly, allowing the first class to experience lower delays, while requirements of high priority classes are still met (*AWRR* row in table 1).

5. Final remarks

WRR is a simple static scheduling algorithm, which does not react to any changes in traffic and system characteristics. On the other hand, priority scheduling (*PRIO*) can be treated as degenerated adaptive *WRR*, which assigns only binary weights to traffic classes, basing on classes priorities and queues lengths. *AWRR* is a fully adaptive scheduling algorithm, which responds to any changes in the serviced traffic.

Results of performed simulations show, that estimation of traffic characteristics and utilization of gathered knowledge in the process of packets scheduling may significantly improve (up to 30% for first traffic class) the level of delivered quality of services.

Presented approach to adaptive packet scheduling is based on adaptation through identification methodology. Identification refers to prediction of future *QoS* parameters of serviced traffic, basing on values of parameters of primary scheduling algorithm. Adaptation relies on choosing new parameter of primary scheduling algorithm, which minimize certain quality of service index and deliver required level of *QoS* for actual traffic characteristics.

The choice of connection delays as the *QoS* parameters and *WRR* as the primary scheduling algorithm was based on the simplicity of implementation of proposed solutions in the simulator. Presented adaptive scheduling approach, however, can be applied for arbitrary *QoS* measure and scheduling methodology. The only difference for other *QoS* measures and scheduling algorithms is the assumed model Φ of predicted *QoS* parameters. For example, vector $\mathbf{q}(n)$ of delivered *QoS* may describe average delay and jitter for each traffic class. In such a case vector $\mathbf{q}(n)$ would consist of $2K$ elements.

In the future work, the performance of presented approach should be evaluated for different measures of *QoS* (e.g.: maximal delay, jitter, packet loss ratio, etc.). Moreover, it should be compared, to other commonly used scheduling algorithms. Interesting results may be obtained by evaluation of the *QoS* level delivered by proposed approach in systems with admission control mechanisms.

Acknowledgements

The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.01.02-00-045/09.

Bibliography

[1] T. Czachórski. *Queuing models in estimation efficiency for computer systems and networks*, PKJS, Gliwice, 1999. (In Polish).

[2] J. Drapała and J. Świątek. Backpropagation through time for learning of interconnected neural networks - identification of complex systems. In *Proceedings of the 19th International Conference on Systems Engineering*, Las Vegas, Nevada, USA, 2008.

[3] S. Floyd and V. Jacobson. Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3:365–386, 1995.

[4] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, New York, 2001.

[5] A. Grzech. *Teletraffic control in the computer communication networks*, Wroclaw University of Technology Publishing House, 2002. (In Polish).

[6] A. Grzech, P. Świątek . *Parallel processing of connection streams in nodes of packet-switched computer communication systems*, Cybernetics and Systems, vol. 39, iss. 2, pp. 155-170, 2008.

[7] A. Grzech, P. Świątek. Modeling and optimization of complex services in service-based systems. *Cybernetics and Systems*. 2009, vol. 40, s. 706–723

[8] A. Grzech, P. Świątek. The influence of load prediction methods on the quality of service of connections in the multiprocessor environment. *Systems Science*, 2009, vol.35, no. 3, pp. 7-14.

[9] C.C. Ku and K.Y. Lee. *Diagonal Recurrent Neural Networks for Dynamic Systems Control*. IEEE Trans. on Neural Networks, Vol. 6, No. 1, January 1995. 144–155.

[10] P. Świątek. Modeling of multi-queue-single-processor queuing system for load balancing purposes. In *Knowledge processing and reasoning for information society*, pages 249–262. Exit, Warsaw, Poland, 2008.

[11] P. Świątek. Evaluation of on-line algorithms for packet scheduling in a network node. In *Proceedings of the 12th IEEE International Conference on Methods and Models in Automation and Robotics*, pages 1129–1132, Międzyzdroje, Poland 2006.

[12] P. Świątek. Multistage Packet Processing in Nodes of Packet-Switched Computer Communication Networks. *Theoretical and Applied Informatics*, 2007, vol. 19, no. 4, pp. 267-279