Twostage Packet Scheduling in the Network Node

Adam Grzech * **Paweł Świątek** ** **Piotr Rygielski** ***

* *Institute of Informatics, Wrocław University of Technology,*
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
(e-mail: adam.grzech@pwr.wroc.pl)
** *Institute of Informatics, Wrocław University of Technology,*
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
(e-mail: pawel.swiatek@pwr.wroc.pl)
*** *Institute of Informatics, Wrocław University of Technology,*
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
(e-mail: piotr.rygielski@pwr.wroc.pl)

Abstract: In this paper the problem of twostage packet scheduling on parallel processors is considered. It is assumed, that each processor schedules packets according to *Weighted Round Robin* (*WRR*) rule. In order to deliver required level of the quality of service (*QoS*) weights of local *WRR* algorithms are adapted such that *QoS* requirements are met for all distinguished traffic classes. Adaptation of *WRR* weights relies on the adaptation through identification (Bubnicki, 2005) methodology with the diagonal recurrent neural network (*DRNN*) applied as the model of *QoS* parameters. Simulation study of the performance of proposed scheduling methodology shows, that it can deliver better *QoS* guaranties and higher systems throughput than commonly used reference algorithms.

Keywords: Packet scheduling, adaptation, diagonal recurrent neural network, QoS.

1. INTRODUCTION

One of the most important mechanisms for delivering the quality of services (*QoS*) in computer communication packet-switched networks is packet scheduling in network nodes (Czachórski, 1999). Delivering *QoS* consists of guarantying for each separate stream of packets (e.g. connection) certain values of communication parameters, such as: maximum or average packet delay, jitter, packet loss ratio, etc (Grzech, 2002). Required values of communication parameters depend on the traffic class, to which separate streams of packets belong to. Traffic classes are often distinguished basing on the applications, which generate the traffic. The task of packet scheduling algorithm is to service packets belonging to different streams in such a way, that *QoS* requirements are met for each separate stream of packet (Świątek, 2007).

In this paper the problem of twostage packet scheduling on parallel processors is considered. It is assumed, that each new stream of packets incoming to the network node is directed for service to one of certain number of parallel processors. Next, traffic processed on parallel processors is aggregated in the output interface and forwarded to the network (Grzech and Świątek, 2008). In order to guarantee required level of *QoS* packets belonging to separate streams must be properly scheduled on parallel processors (first stage) and in the output interface (second stage).

Here, it is assumed, that each parallel processor in the first stage and output interface in the second stage have their own local scheduling algorithms. Delivered level of *QoS* depends on the parameters of local scheduling algorithms and parameters of serviced traffic (e.g. packet intensity, queues lengths, etc.). Thus, by introducing certain model Φ of the delivered level of *QoS*, the problem of twostage packet scheduling can be reduced to the problem of adaptation through identification of local scheduling algorithms.

In this paper it is assumed, that packets in both stages are scheduled according to *Weighted Round Robin* (*WRR*) algorithm and that the delivered *QoS* is modeled by *diagonal recurrent neural network* (*DRNN*).

The paper is organized as follows. In section 2 assumed model of serviced traffic is described. In section 3 the model of the considered network node with multistage parallel processing is presented. The problem of twostage packet scheduling is formulated in section 4, while adaptive twostage packet scheduling algorithm is given in section 5. Exemplary results of performed simulation study are presented in section 6. Finally, in section 7 conclusions are drawn and directions of future work are presented.

2. TRAFFIC MODEL

For the purpose of this paper, it is assumed, that the aggregated stream of packets incoming into the network node is composed of substreams of packets characterized by the same source and destination addresses (Grzech, 2002). Throughout the paper, such a substream of packets, is referred to as connection. Each connection belongs to one of certain number (say K) of distinguished traffic classes. Each j-th connection c_j can be characterized by: class number k_j, arrival time t_j, duration τ_j and the sequence of arrival times of packets belonging to this connection $\{t_{j1},\dots,t_{ji_j}\}$ (see fig. 1).

$$c_j = < k_j, t_j, \tau_j, \{t_{j1},\dots,t_{ji_j}\} >, \tag{1}$$

where $k_j \in \{1,\dots,K\}$ and i_j is the number of packets within connection c_j.

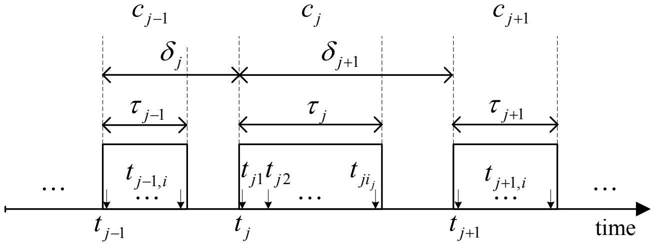

Fig. 1. Model of connections from k-th traffic class.

It is also assumed, that parameters characterizing connections from the same traffic class k are realizations of random variables described by the same probability distributions. Therefore, each traffic class $k \in \{1,\dots,K\}$ can be characterized by three probability distribution 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_j between arrival of two consecutive connections (c_{j-1} and c_j), duration τ_j of connection and time interval α_{ji} between arrival of two consecutive packets within single connection. Vectors $\boldsymbol{\delta}_k$, $\boldsymbol{\tau}_k$ and $\boldsymbol{\alpha}_k$ are parameters of corresponding distribution functions.

3. MODEL OF THE NETWORK NODE

In order to improve throughput and reliability of network nodes they are often equipped with multiple parallel processors, which perform specific operations on packet streams incoming to the node. Processing of connection in such nodes consists of three stages (see fig. 2). On the first stage each new arriving connection is directed for processing to one of certain number (say M) of parallel processors $\boldsymbol{P}_m$, $m = 1,\dots,M$. On the second stage connections are processed on parallel processors. Operations performed on connections on the second stage depend on the traffic class to which each connection belongs to. On the third stage processed traffic is merged by scheduler $\boldsymbol{S}$ into single stream of packets and forwarded to the network.

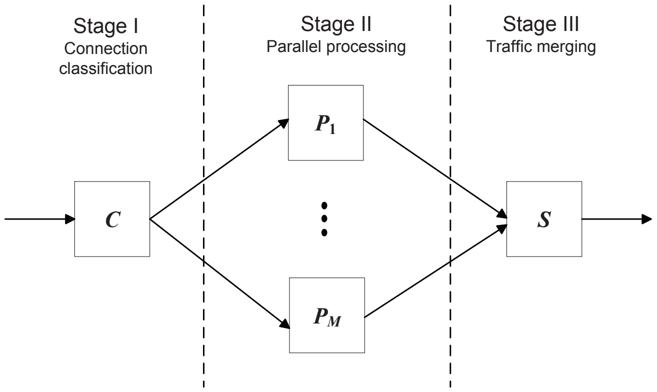

Fig. 2. Three stages of connection processing on parallel processors.

Each of parallel processors $\boldsymbol{P}_m$ and scheduler $\boldsymbol{S}$ can be treated as multi-queue single-processor queueing system presented on figure 3, where vectors $\mathbf{z}_m(n)$, $\mathbf{x}_m(n)$ and $\mathbf{v}_m(n)$ denote respectively: number of packets from each traffic class incoming to processor in n-th time interval, number of packets waiting for processing at the beginning of n-th time interval and number of packets from each traffic class serviced by processor during n-th time interval (Świątek, 2008).

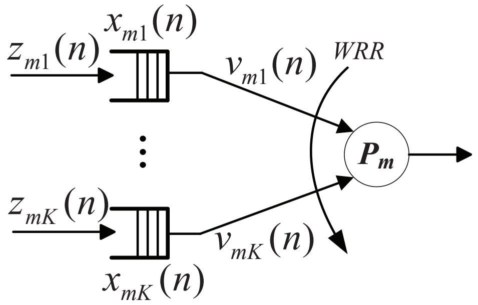

Fig. 3. Network processor as the multi-queue single-processor queueing system.

It is assumed, that packets on processors are scheduled according to *Weighted Round Robin* (*WRR*) algorithm, where each processor has its own weights $\mathbf{w}_m(n)$, which are changed in each time step n. Moreover, each processor $\boldsymbol{P}_m$ and scheduler $\boldsymbol{S}$ are characterized by processing speed μ_m. Therefore, number of packets $\mathbf{v}_m(n)$ serviced on each processor in each time step n can be calculated as (Bertsekas and Gallager, 1991):

$$\mathbf{v}_m(n) = \mu_m \mathbf{w}_m(n). \tag{2}$$

State of the whole system can be described by three vectors $\mathbf{z}(n)$, $\mathbf{x}(n)$ and $\mathbf{w}(n)$, which are concatenations of vectors describing each parallel processor and scheduler $\boldsymbol{S}$:

$$\mathbf{z}(n) = [\mathbf{z}_1(n),\dots,\mathbf{z}_M(n),\mathbf{z}_S(n)]$$
$$\mathbf{x}(n) = [\mathbf{x}_1(n),\dots,\mathbf{x}_M(n),\mathbf{x}_S(n)]$$
$$\mathbf{w}(n) = [\mathbf{w}_1(n),\dots,\mathbf{w}_M(n),\mathbf{w}_S(n)]$$

Denote by $\mathbf{q}(n) = [q_1(n),\dots,q_K(n)]$ a vector of the quality of service indices of connections from each traffic class $k = 1,\dots,K$. Quality of service can be measured as average or maximal connection delay, packet loss ratio, etc. In general, *QoS* indices $\mathbf{q}(n)$ depend on the current state of system:

$$\mathbf{q}(n) = f\left(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n)\right), \tag{3}$$

where function f is in general unknown.

In this work it is assumed, that quality of services $\mathbf{q}(n)$ is measured as the average delay of connections from each traffic class. Nevertheless, it is possible to characterize quality of service of different traffic classes by different number and kind of parameters. In such a case, vector $\mathbf{q}(n)$ will have more than K elements. As an example consider a case, where $K = 2$ traffic classes are distinguished: regular traffic and internet telephony (*VoIP*). In this case

QoS vector $\mathbf{q}(n)$ may have $\kappa = 4$ elements. First one representing average delay of regular traffic and remaining three representing average and maximal delay and jitter of packets in *VoIP* connections.

4. PROBLEM STATEMENT

The main objective of the traffic flow control algorithm in a network node is to maximize systems throughput and to guarantee required level of quality of services. Throughput maximization is equivalent to minimization of average traffic delay. Required level of *QoS* is often stated in terms of constraints on relevant quality parameters of serviced traffic.

Denote by $\overline{\mathbf{q}}$ the vector of requirements concerning respective quality parameters $\mathbf{q}(n)$ for all traffic classes. Given certain criterion function $Q(\mathbf{q}(n))$, which calculates average traffic delay, the problem of twostage packet scheduling can be formulated as follows.

Given:

- state vectors $\mathbf{z}(n)$ and $\mathbf{x}(n)$,
- vector of processors speeds $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M, \mu_S]$,
- function f for calculation of *QoS* parameters (3),
- vector of *QoS* requirements $\overline{\mathbf{q}}$,
- criterion function Q.

Find: Such a vector $\mathbf{w}^*(n)$ of local scheduling algorithms weights, for which average traffic delay is minimized and quality of service requirements are satisfies:

$$\mathbf{w}^*(n) = \arg\min_{\mathbf{w}} Q(\mathbf{q}(n)) \tag{4}$$

with respect to:

$$\mathbf{q}(n) \leq \overline{\mathbf{q}}. \tag{5}$$

Since $\mathbf{q}(n) = f\left(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n)\right)$ twostage packet scheduling problem can be rewritten as follows:

$$\mathbf{w}^*(n) = \arg\min_{\mathbf{w}} Q\big(f\left(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n)\right)\big) \tag{6}$$

with respect to:

$$f\left(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n)\right) \leq \overline{\mathbf{q}}. \tag{7}$$

Depending on the form of functions f and Q above optimization problem can be solved by different methods (Ikonen and Najim, 2002). In general, however, this problem belongs to the class of NP-hard problems. Even if function f (describing the influence of systems state on quality of service) was known, application of exact optimization methods would be useless, due to the fact of high computational complexity of stated problem and near real-time constraints on solving time.

5. TWOSTAGE PACKET SCHEDULING ALGORITHM

In order to find good approximation of optimal solution of problem (6, 7) in an acceptable time problem decomposition must be applied. Here, we utilized adaptation through identification approach [], which consists of two steps performed iteratively in consecutive time intervals $n = 1, 2, \ldots$. In the first step unknown function $f\left(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n)\right)$ is approximated by certain assumed model $\Phi(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n); \boldsymbol{\theta}(n))$, where $\boldsymbol{\theta}(n)$ is the vector of models parameters. In the second step following optimization task, which arises from problem (6, 7) after substitution of function f by model $\Phi(\boldsymbol{\theta}(n))$, is solved:

$$\mathbf{w}^*(n) = \arg\min_{\mathbf{w}} Q\big(\Phi(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n); \boldsymbol{\theta}^*(n))\big) \tag{8}$$

with respect to:

$$\Phi(\mathbf{z}(n), \mathbf{x}(n), \mathbf{w}(n); \boldsymbol{\theta}^*(n)) \leq \overline{\mathbf{q}}, \tag{9}$$

where $\boldsymbol{\theta}^*(n)$ is a vector of optimal model parameters found in previous approximation step.

Presented above two step process of adaptation of weights of local *WRR* algorithms through identification of the influence of systems state on delivered quality of service constitutes *Adaptive Weighted Round Robin* (*AWRR*) algorithm.

In the rest of this paper it is assumed, that the model $\Phi(\boldsymbol{\theta}(n))$ of function f is a *Diagonal Recurrent Neural Network* (*DRNN*). Therefore, the approximation of f by model $\Phi(\boldsymbol{\theta}(n))$ is performed according to *Backpropagation Through Time* (*BPTT*) method (Drapała, 2007).

In order to find optimal values of *WRR* weights $\mathbf{w}^*(n)$, which minimize the quality of service index (8) with respect to *QoS* constraints (9), *simulated annealing* metaheuristic was applied (Rutkowski, 2006). However, for the cases, when the number$M$ and K of parallel processors and distinguished traffic classes is not high ($M \leq 4$ and $K \leq 4$), exhaustive search may be used.

6. SIMULATION STUDY

In order to evaluate the quality of service delivered by proposed AWRR algorithm, it was compared to reference algorithms: WRR and PRIO. WRR is the weighted round robin with weights chosen to reflect traffic class priorities. PRIO is the priority scheduling algorithm, which allocates all system resources to class, with has the highest priority.

In the simulations, it was assumed, that there are $K = 2$ traffic classes. Connections from each traffic class were generated by *ON/OFF* sources. First traffic class was sending data periodically, and the second one almost continuously. Priorities of traffic classes were set to $p_1 = 5$ and $p_2 = 1$. *QoS* requirements of classes were $\overline{q}_1 = 50$ and $\overline{q}_2 = \infty$. Above assumptions mean, that the second traffic class was the *best effort* traffic.

Exemplary results obtained for the AWRR algorithm were presented on figure 4. There is presented average delay of connections from each traffic class during the simulation period. Analysing the chart, it is easy to notice, that the requirements of amount of delay for the first traffic class were met. Moreover, the AWRR algorithm has allocated only such amount of resources, that were necessary. Remaining resources were allocated to the second traffic class allowing to achieve lower delays. In the periods when first traffic class was not sending data, all resources were allocated to the second traffic class, and the delay was near zero. When there is traffic from first class present in system, the resources are reallocated basing on priority values and the second traffic class connections are being serviced much slower.

On the figure 5 average delays of connections from the second traffic class under control of compared algorithms

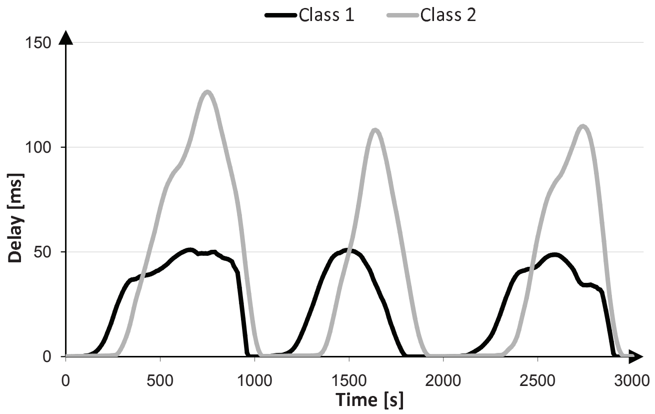

Fig. 4. Delay of connection from two traffic classes for the *AWRR* algorithm.

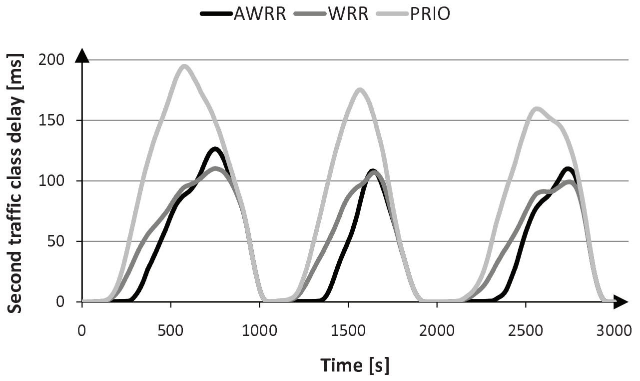

Fig. 5. Delay of connections from second traffic class (best effort traffic) for three considered scheduling algorithms: *AWRR*, *WRR* and *PRIO*.

were presented. One can notice, that the lowest delays are obtained for the proposed AWRR algorithm. The WRR algorithm delivered higher average delay, and the PRIO the highest. Detailed data about the delay of second traffic class during simulation were presented in table 1.

Algorithm	Class 2 avg. delay	Class 2 max delay
PRIO	77,53	194, 69
WRR	47,8	110, 27
AWRR	39,84	126, 4

Table 1. Average and maximum delay for *best effort* traffic class delivered by compared algorithms: *AWRR*, *WRR* and *PRIO*

As we can see, the proposed AWRR algorithm delivered lowest average delay for the second traffic class. There were short moments, when the WRR delivered lower delays than AWRR, but it does not have large influence on overall average delay value. Moreover, the AWRR algorithm was able to deliver delay values near to zero for the longest period of time.

7. FINAL REMARKS

In this paper the problem of twostage packet scheduling in a network node was considered. It was assumed, that each of parallel processors and scheduler service incoming packets according to *WRR* algorithm. The task was to find such a vector of local scheduling algorithms weights, that systems throughput is maximized and *QoS* requirements are met. Since originally formulated problem is NP-hard and cannot be directly solved due to lack of knowledge about the influence of systems state on delivered level of quality of services, problem decomposition was performed.

Approximation of optimal solution of twostage packet processing problem was found according to adaptation through identification methodology. Proposed solution method (*AWRR*) consists of two iteratively repeated steps: identification of systems behaviour with recurrent neural network applied as systems model and optimization of predicted by model *QoS* criterion.

Future research will be focused on application of proposed scheduling methodology in systems, where quality of service of different traffic classes is characterized by different number and kind of parameters. This may also involve application of other models for prediction of future *QoS* parameters.

ACKNOWLEDGEMENTS

This work was supported by the Polish Ministry of Science and Higher Education under Grant No. N N 516 3838 34 (2008-2009).

REFERENCES

Bertsekas, D. and Gallager, R. (1991). *Data networks.* Prentice Hall, London, UK.

Bubnicki, Z. (2005). *Modern control theory.* Springer, Berlin, Germany.

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

Drapała, J. (2007). On convergence of algorithm of neural networks learning for global modeling of complex systems. In *Proceedings of the 16th International Conference on Systems Science*, 359–368. Wroclaw, Poland.

Grzech, A. (2002). *Control of traffic in computer communication networks.* Wroclaw University of Technology Publishing House, Wroclaw, Poland. (in Polish).

Grzech, A. and Świątek, P. (2008). On the problem of traffic load balancing in the network node. In *Proceedings of the 7th Polish - British Workshop on Computer Systems Engineering*, 212–223. Kudowa Zdrój, Poland.

Ikonen, E. and Najim, K. (2002). *Advanced Process Identification and Control (Control Engineering).* CRC Press, New York.

Rutkowski, L. (2006). *Methods and Techniques of Artificial Intelligence.* PWN, Warsaw, Poland.

Świątek, P. (2007). Multistage packet processing in nodes of packet-switched computer communication networks. *Theoretical and Applied Informatics*, 19(4).

Świątek, P. (2008). Modeling of multi-queue-single-processor queuing system for load balancing purposes. In N.N. et al. (ed.), *Knowledge processing and reasoning for information society*, 249–262. Exit, Warsaw, Poland.