Multistage Packet Processing in Nodes of Packet-Switched Computer Communication Networks

Pawel Świątek

Institute of Information Science and Engineering
Wrocław University of Technology
pawel.swiatek@pwr.wroc.pl


Abstract: In this paper the model of the multistage packet processing system is defined in terms of the control theory. Basing on the proposed model, problems of single and multistage packet scheduling and coordination of local control (scheduling) algorithms in the multistage scheduling system are defined. Since formulated problems are NP-hard, optimal algorithms cannot be applied to the real-time traffic control. Instead, one can use on-line approaches, which approximate the optimal solution. Therefore, we show how to adapt existing single stage scheduling algorithms to the on-line version of the multistage packet scheduling problem. Finally, we give some remarks concerning the application of the artificial intelligence methods to the considered problem. The discussion is followed by the illustrative numerical example, which confirms the efficiency of proposed solutions.

Keywords: Packet scheduling, traffic flow control, Quality of Service

1. Introduction

Providing quality of service (QoS) in computer communication packet-switched networks is strongly related to the problem of traffic flow control in the network node. Efficient traffic flow control requires that the aggregated stream of packets incoming into the network node is decomposed into separate traffic classes, which require different types of service [6]. Depending on the traffic class there are different traffic parameters, that are taken into account during the evaluation of the quality of service e.g.: jitter for voice transmission, delay for real-time systems and packet loss ratio for data transfer. Additional issues of QoS management include: traffic shaping, link sharing and fairness.
In this paper a new model of multistage packet processing system is proposed. Processing of the incoming stream of packets consists of three stages (Fig. 1). First, aggregated stream of packets is divided into substreams (e.g., connections) and each substream is assigned for further processing to one of \( M \) parallel processors \( O_m, m \in \{1, ..., M\} \) on the second stage. Finally, output of the second stage processing is aggregated again into single stream and forwarded to the network by device \( O_{M+1} \).

![Fig. 1. Multistage packet scheduling system](image)

Even though there exist a number of models and algorithms for single stage packet processing, the proposed approach has some advantages. First of all, parallel processing significantly increases systems throughput and reliability. Moreover, in such a system there is a possibility to make use of specialized devices adapted to specific features of certain packet substreams (traffic classes), and thus able to perform certain task better (faster, more accurate) than universal devices.

In this paper, basing on the control theory, problems of single- and multistage packet scheduling in the network node are defined. Since formulated problems are NP-hard, optimal algorithms cannot be applied to real-time traffic control. Instead, one can use on-line approaches, which approximate the optimal solution. Therefore, we show how to adapt existing single stage scheduling algorithms to the on-line version of multistage packet scheduling problem. Finally, we give some remarks concerning the application of artificial intelligence methods to the considered problem.

### 2. Single stage packet processing

Generally, traffic flow control in the network node consist in scheduling packets from different traffic classes. Let \( K \) be the number of traffic classes. Packets belonging to each traffic class \( k \in \{1, \ldots, K\} \) flow into the node according to certain probability distribution with the mean intensity \( \lambda^{(k)} \) and wait for service in \( k \)-th queue (Fig. 2).
Moreover, each class $k$ is characterized by priority $p^{(k)}$ and quality of service criterion $q^{(k)}$. Let $p = [p^{(1)}, \ldots, p^{(K)}]$ and $q = [q^{(1)}, \ldots, q^{(K)}]$ are vectors of priorities and QoS criteria respectively.

The task of decision algorithm is to schedule packets from different classes (queues) such that quality of service guarantees are satisfied, i.e. certain criterion $Q(q, p)$ is minimized.

Foregoing packet scheduling system can be modelled as the input-output control plant $CP$ [3] (Fig. 3).

Let $u_n = [u_n^{(1)} \ldots u_n^{(K)}]^T$ (where $n$ is the number of control step) be the input (decision) vector of the control plant, where $u_n^{(k)} = 1$ if the packet, which is going to be serviced belongs to $k$-th class and $u_n^{(i)} = 0$ for $i \in \{1, \ldots, K\}\{k\}$. Moreover, let intensities of each traffic class are the disturbance $z_n = [\lambda_n^{(1)} \ldots \lambda_n^{(K)}]^T$. Finally, let the vector of temporary values of the quality of service criteria of each class is the output of the system $y_n = [q_n^{(1)} \ldots q_n^{(K)}]^T$. Since the state $x_n = [x_n^{(1)} \ldots x_n^{(K)}]^T$ of the control plant $CP$ is precisely defined by lengths of queues $k$, then the control plant $CP$ can be described by the following state equation

$$
\begin{align*}
    x_{n+1} &= x_n - u_n + z_n \\
    y_n &= F(x_n, u_n, z_n)
\end{align*}
$$

where function $F$ is (in general) unknown, however values of $y_n$ can be measured. The quality of control (service) $Q_n$ is evaluated according to certain criterion function $\varphi$, and $Q_n = \varphi(y_n, p)$. 

Fig. 2. Single stage packet scheduling

Fig. 3. Model of the single stage packet scheduling system
For such a system we formulate the following control (decision making) problem.

**Problem 1 (P1):**

*Given:* $\phi, F, p, x_0, N$

*Find:* The sequence of optimal decisions $(u^*_0, \ldots, u^*_{N-1})$ such that quality criterion function $\sum_{n=1}^{N} Q_n$ is minimized:

$$
(u^*_0, \ldots, u^*_{N-1}) = \arg \min_{u_0, \ldots, u_{N-1}} \sum_{n=1}^{N} Q_n
$$

(2)

Since

$$
Q_n = \varphi(y_n, p) = \varphi(F(x_n, u_n, z_n), p) = \Phi(x_n, u_n, z_n, p)
$$

(3)

then (2) reduces to

$$
(u^*_0, \ldots, u^*_{N-1}) = \arg \min_{u_0, \ldots, u_{N-1}} \sum_{n=1}^{N} \Phi(x_n, u_n, z_n, p)
$$

(4)

There is possibility to find optimal solution of some special cases of problem P1 by applying dynamic programming procedure, in general however, this problem belongs to the class of NP-hard problems and it is highly unlikely to solve it in acceptable time. The problem becomes more complex as we note, that function $F$ (and in consequence function $\Phi$) is unknown and disturbances $z_n$ (which are the intensities of traffic classes) are random. Therefore, in practice, only heuristic on-line algorithms are applied, which approximate optimal solution by minimizing the temporary quality of service criterion $Q_n$.

In literature, there exist a number of efficient algorithms for the above-mentioned single-stage packet scheduling problem. Each approach can be classified to one of variety of scheduling methodologies, which differ in the model used to describe the required and delivered quality of services. The simplest method – *Weighted Round Robin* – belongs to the wider group of algorithms named *Class Based Queuing* [5]. Next, in [12] author presents commonly used link sharing algorithm – *Token Bucket* and its hierarchical version – *Hierarchical Token Bucket*. Another approach to traffic flow control are methods based on the fairness principle (e.g.: *Weighted Fair Queuing* [8], *Worst-case Fair Weighted Fair Queuing* [1], *Hierarchical Packet Fair Queuing* [2]). Finally, a significant improvement in the quality of packet scheduling was achieved after introducing new scheduling approach based on service curves (i.e.: *Fair Service Curve* [7] and *Hierarchical Fair Service Curve* [9]).

Operation of the single-stage packet scheduling system under the control of arbitrary algorithm ($H$) can be modelled as the input-output control system depicted on Fig. 4.

The algorithms proposed in the literature yield worse or better approximations of optimal solutions. In order to enhance the performance of those methods, their parameters
are often adapted to match the conditions prevailing in the systems environment. Such a control system, where decisions made by the scheduling algorithm $H$ are improved by certain adaptation algorithm $A$, is depicted on Fig. 5. The adaptation algorithm $A$ changes some parameters $a_i$ of the scheduler $H$ basing on the knowledge about the systems environment collected by measuring the input and output signals of the system.

![Fig. 4. Single-stage packet scheduling as the input-output control system](image1)

![Fig. 5. General idea of adaptation in the single-stage packet scheduling system](image2)

![Fig. 6. Model of the single-stage packet scheduling system controlled by the AWRR algorithm](image3)

Depending on the complexity of the scheduling algorithm $H$ and the way its parameters are changed (adapted), different quantities are taken into account during the process of control and adaptation. For example, an adaptive weighted round robin (AWRR) with weights being changed according to the reinforcement learning approach is proposed in [11]. In this case (Fig. 6), the scheduling algorithm $H$ does not need any input information for the decision making (open control system) and the adaptation algorithm $A$ uses only measured values of the output of the control plant, i.e. temporary values of the quality of service criteria of each traffic class.
3. Multistage packet processing

Consider the multistage packet scheduling system from Fig. 1, where each object \( O_m, m \in \{1, \ldots, M + 1\} \) has the structure as shown on Fig. 2.

The traffic incoming into the network node is divided into substreams (e.g. connections) and directed for further processing by dispatcher \( O_0 \). Next, parallel substreams are processed separately on devices \( O_m, m \in \{1, \ldots, M\} \) and finally merged into one output stream on processor \( O_{M+1} \).

The task of such a system is to schedule packets in such a way, that certain global quality of service criterion is minimized.

On Fig. 7 the multistage packet scheduling system as the complex input-output control system is presented.

Let \( X_n = [x_{1,n} \ldots x_{M+1,n}]^T \) and \( U_n = [u_{1,n} \ldots u_{M+1,n}]^T \) are respectively the state and decision vectors of devices \( O_m, m \in \{1, \ldots, M + 1\} \) and \( Z_n = [z_{01,n} \ldots z_{0M,n} z_{1M+1,n} \ldots z_{MM+1,n}]^T \) is the vector of disturbances. Outputs \( y_{m,n} \) are vectors of temporary values of local quality of service of each traffic class, and \( y_n \) is the vector of temporary values of quality of service of each class on the output of the whole system. The global quality of service \( Q_n \) is calculated according to certain criterion function \( Q_n = \varphi(y_n, p) \).

The problem of packet scheduling in such a system can be formulated as follows.

**Problem 2 (P2):**

**Given:** \( \varphi, F, p, X_0, N \)

**Find:** The sequence of optimal decisions \( (U_0^*, \ldots, U_{N-1}^*) \) such that quality criterion function \( \sum_{n=1}^{N} Q_n \) is minimized:
(U_0^*, \ldots, U_{N-1}^*) = \arg \min_{U_0, \ldots, U_{N-1}} \sum_{n=1}^{N} Q_n \tag{5}

what after transformation analogical to (3) gives

(U_0^*, \ldots, U_{N-1}^*) = \arg \min_{U_0, \ldots, U_{N-1}} \sum_{n=1}^{N} \Phi(X_n, U_n, Z_m, p) \tag{6}

Note, that the difference between single- and multistage packet scheduling is that decisions made on the second stage (parallel processing) influence the quality of service on the third stage. In fact, single stage system from Fig. 2 is a part of the multistage system from Fig. 1 and in consequence problem P1 is a subproblem of problem P2.

Obviously, for the same reasons as for the single-stage case, determination of solution (6) is not possible and one should apply on-line algorithms, which minimize the temporary quality of service criterion Q_n in successive control steps.

4. Improving the performance of the system

Since each of devices O_m, m \in \{1, \ldots, M + 1\} can be treated as the single stage scheduling system, it is possible to apply known packet scheduling strategies as the local control algorithms in the multistage system. Note, however, that such an approach can yield only locally optimal solutions, which in general are worse from the global point of view [10]. Only for simple cases and special forms of the criterion function it can be shown, that locally and globally optimal solutions are the same. On the other hand, calculating global solution may be too complex (time-consuming) to be applied in real-time control systems.

In order to improve performance of the system an upper level control algorithm may be used, which would coordinate [4] local scheduling algorithms H_m. The task of such a coordinator would be to calculate new parameters of decision algorithms basing on measured values of systems inputs, outputs and states. In such a case the coordinator acts as an adaptator.

Proposed multistage packet processing system, which includes adaptation (block A) is presented on Fig. 8. Due to clarity reasons a substitute plants OZ_m (depicted on Fig. 9) were introduced.

Let l be the adaptation step number, which lasts N basic control steps. Moreover, let C_l = [c_{l,N+1} \ldots c_{lN}] be the matrix of systems parameters during the l-th adaptation step, where vectors c_n = [c_{1,n} \ldots c_{M+1,n}]T describe the system in the n-th control step and n = (l - 1)N + 1, \ldots, lN. Additionally, denote by a_l = [a_{1,l} \ldots a_{M+1,l}]T the vector of new parameters of local control (scheduling) algorithms H_m, m = 1, \ldots, M + 1.
The task of adaptation consists in that for the sequence of measured systems parameters $C_l$ one should find such a new vector of control algorithm parameters $a_l$ that minimizes the value of the global quality of service criterion $\sum_{n=(l-1)N+1}^{lN} Q_n = \sum_{n=(l-1)N+1}^{lN} \Theta(c_n, a_l) \tilde{\Theta}_N(C_l, a_l)$ during $N$ control steps.

For such described system the following problem of adaptation (coordination) can be formulated.

**Problem 3 (P3):**

**Given:** $\Theta$, $N$, $C_l$

**Find:** The vector of optimal parameters $a_l^*$ such that quality criterion function $\sum_{n=(l-1)N+1}^{lN} Q_n = \sum_{n=(l-1)N+1}^{lN} \Theta(c_n, a_l) \tilde{\Theta}_N(C_l, a_l)$ is minimized:

$$a_l^* = \arg \min_{a_l} \tilde{\Theta}_N(C_l, a_l)$$  \hspace{1cm} (7)
Unfortunately, function $\Theta$, and what follows, function $\bar{\Theta}$ are not known. It is merely possible to measure consecutive values of criterion $Q_n$ for varying parameters of the system. Thus, analytical solution of the adaptation problem $P3$ cannot be found.

There are, however, another two approaches to handle that problem. In the first method we assume certain function $Q_n = \Theta(c_n, a_l; b)$ and identify its parameters $b$ during the run of the system, what allows us to solve problem (7) by applying the successive approximations method:

$$a_{l+1} = a_l - K \cdot w_l$$  \hspace{1cm} (8)

where

$$w_l = \nabla_a \bar{\Theta}(C_l, a_l; b_l)\big|_{a=a_l}$$  \hspace{1cm} (9)

and $b_l$ is a vector of identified parameters of the assumed model $\Theta$.

Such an approach, which treats the whole system as the “black box” may sometimes yield questionable results, because assumed model not always allows us to take into consideration the dynamics of the system. Therefore, the choice of the model $\Theta$ has a major impact on systems performance.

Another approach, which is more accurate (and of course computationally more complex) takes advantage of the fact, that even though the function $\Theta$ is not known, we have the description of the system in the form of applied algorithms. Thus, information about the temporary state of the system $c_n$ allows us to calculate the value of the criterion function $Q_n = \Theta(c_n, a)$ for arbitrary values of parameters $a$ by means of the computer simulation $S$.

$$Q_n = S(c_n, a)$$  \hspace{1cm} (10)

In this case we can again apply the extremal control algorithm (8), but the $i$-th component of the vector $w_l$ is calculated according to the trial steps method

$$w_l^{(i)} = \frac{S(c_n, a_l - \delta_i) - S(c_n, a_l + \delta_i)}{2\sigma_i}$$  \hspace{1cm} (11)

where $\delta_i$ is a zero vector except the $i$-th component equal to $\sigma_i$ (the trial step value).

Provided we have a fast enough systems simulator, it is also possible to make use of any of numerical optimization methods (including artificial intelligence - e.g.: reinforcement learning, genetic algorithms) to calculate optimal values of parameters $a^*_l$ for given systems parameters $C_l$ by use of the following algorithm

$$a^*_l = \arg \min_a S_N(C_l, a).$$  \hspace{1cm} (12)

Finally, the simulator of the system may be applied as the training algorithm for the expert system $ES$, which would return optimal values of $a^*_l$ for sequentially measured systems parameters $C_l$

$$a^*_l = ES(C_l).$$  \hspace{1cm} (13)
In this case, simulator is the trainer, and the learning expert may be based on one of the following approaches: knowledge base with a set of rules, neural network, or another expert with knowledge representation (e.g.: uncertain variables, fuzzy sets). Additionally, the trained expert may validate and update its knowledge during the run of the system [3].

The crucial issue, which must be taken into consideration during the process of design of the adaptation system is the length of the adaptation step. It must be long enough to allow the computationally complex adaptation algorithm to execute. On the other hand, if the step is too long, systems conditions to which we are trying to adapt will change.

5. Simulation study

In order to show the difference between presented scheduling schemes a simple experimental example is provided. In this example three scheduling algorithms are compared: Weighted Round Robin – WRR and Adaptive Weighted Round Robin – AWRR and AWRR2. Adaptation in AWRR and AWRR2 is based on the reinforcement learning approach introduced in [11]. Each of algorithms represents one of scheduling schemes presented in previous sections.

WRR is a simple method, where no adaptation is involved and weights are assigned to classes statically according to the following formula

\[ w^{(k)} = p^{(k)} \left( \sum_{i=1}^{K} p^{(i)} \right)^{-1} \]  

(14)

where \( w^{(k)} \) and \( p^{(k)} \) are \( k \)-th traffic class weight and priority respectively.

In AWRR each algorithm \( H_m, m \in \{1, \ldots, M + 1\} \) calculates new weights \( w^{(k)}_{m,n} \) basing on its local temporary quality of service criterion \( y^{(k)}_{m,n} \) measured at the \( n \)-th control step and the previous value of the weight \( w^{(k)}_{m,n-1} \)

\[ w^{(k)}_{m,n} = g \left( w^{(k)}_{m,n-1}, y^{(k)}_{m,n} \right) \]  

(15)

AWRR2 algorithm, which is an example for coordinated adaptation scheme is similar to AWRR, it uses, however, additional information concerning the value of the global quality of service criterion \( y_n \).

\[ w^{(k)}_{m,n} = g \left( w^{(k)}_{m,n-1}, y_{m,n}, y_{n}^{(k)} \right) \]  

(16)

Presented algorithms were evaluated according to the criterion function defined as follows

\[ Q = \left( \sum_{k=1}^{K} p_k \cdot Q_k \right) \cdot \left( \sum_{k=1}^{K} p_k \right)^{-1} \]  

(17)
where $p_k$ is the $k$-th class priority and $Q_k$ is $k$-th class quality of service criterion defined as

$$ Q_k = \left( \sum_{t=0}^{T} \max\{0, Q_{t,k} - Q^*_k\} \right) \cdot \left( \sum_{t=0}^{T} Q_{t,k} \right)^{-1} $$

where $T$ is the evaluation time horizon and $Q^*_k$ is $k$-th class QoS guarantee.

Since $Q_k$ always satisfies $Q_k \in (0, 1)$, such a function has a natural interpretation as the percentage of overall QoS of traffic class $k$, that violated the quality of service guarantee $Q^*_k$.

On figure 10 the quality of service versus traffic intensity graph for three evaluated algorithms is presented. Simulation results confirmed, that equipping even simple algorithms with additional knowledge about the systems environment and coordinating their operation significantly improves the overall systems performance.

6. Conclusions

In this paper a model of multistage packet scheduling system was introduced. Basing on the model three scheduling problems were formulated. Since defined problems belong to the class of NP-hard problems, exact algorithms cannot be applied in real-time systems. Therefore, it was shown how to adapt existing single stage scheduling algorithms to achieve high performance of the multistage packet processing system. More-
over, the discussion on possible heuristic on-line solution algorithms (including artificial intelligence methods) is provided. Finally, a simple simulation example, that justifies usefulness of proposed methods is given.

Acknowledgments

This work was supported by the Polish State Committee for Scientific Research under Grant No. 3 T11C 029 29 (2005-2007).

References


Wieloetapowe przetwarzanie ruchu teleinformatycznego
w węźle sieci z komutacją pakietów

Streszczenie
Zapewnienie w pakietowych sieciach komputerowych odpowiedniego poziomu bezpieczeństwa komunikacji oraz wysokiej jakości usług transportowych (ang. Quality of Service) związane jest z zarządzaniem ruchem teleinformatycznym w węzłach sieci. Zarządzanie takie polega głównie na szeregowaniu pakietów należących do różnych klas ruchu. Wymaga ono dekompozycji zagregowanego strumienia pakietów wpływającego do węzła na podstrumienie należące do różnych klas ruchu, które to podstrumienie są następnie różnicie obsługiwane w zależności od charakterystyk podstrumienia i wymagań stawianych różnym klasom ruchu.

W literaturze, np.: [1, 2, 5, 7, 8, 9], można znaleźć wiele efektywnych rozwiązań zadania szeregowania pakietów w węzłach sieci teleinformatycznych, jednakże wszystkie te rozwiązania opierają się o jednostanowiskowy model obsługi ruchu. W niniejszej pracy natomiast, proponuje się model wieloetapowej obsługi, w którym przetwarzanie pakietów należących do różnych podstrumieni odbywa się na równoległych stanowiskach obsługi.

Obsługa napływającego ruchu w takim systemie jest podzielona na trzy etapy (rys. 1). Wpływający do węzła strumień pakietów jest rozdzielany na podstrumienie, które z kolei przypisywane są do jednej z K klas ruchu. Następnie urządzenie \( O_0 \) podejmuje decyzję, na którym z M równoległych procesorów \( O_m, m = 1, ..., M \) dany podstrumień ma być obsługiwany. Urządzenie \( O_{M+1} \) ma za zadanie agregować strumienie pakietów wychodzące z procesorów \( O_m, m = 1, ..., M \) w jeden strumień wyjściowy.

Istotą rozważanego problemu polega na tym, że pomimo, iż istnieją efektywne algorytmy szeregowania pakietów na pojedynczych urządzeniach \( O_m \) (aż była traktować jako jednostanowiskowy systemem obsługi), to stosowanie nawet lokalnie optymalnych rozwiązań nie prowadzi na ogół do rozwiązań optymalnych w sensie globalnym [10]. Zatem, aby móc korzystać z niewątpliwych zalet proponowanego modelu, należy dla niego zaprojektować globalnie efektywne algorytmy szeregowania pakietów.