

Article

Hierarchical Queue Management Priority and Balancing Based Method under the Interaction Prediction Principle

Oleksandr Lemeshko ¹, Oleksandra Yeremenko ^{1,*}, Larysa Titarenko ^{1,2} and Alexander Barkalov ^{2,3}

¹ V.V. Popovskyy Department of Infocommunication Engineering, Kharkiv National University of Radio Electronics, 61166 Kharkiv, Ukraine

² Institute of Metrology, Electronics and Computer Science, University of Zielona Góra, ul. Licealna 9, 65-417 Zielona Góra, Poland

³ Department of Computer Science and Information Technology, Vasyl Stus' Donetsk National University, 600-richchia Str. 21, 21021 Vinnytsia, Ukraine

* Correspondence: oleksandra.yeremenko@nure.ua or oleksandra.yeremenko.ua@ieee.org

Abstract: This work is devoted to improving a two-level hierarchical queue management method based on priority and balancing under the interaction prediction principle. The lower level of calculations was connected with the problem optimization solution and was responsible for two tasks. Firstly, the packet flow aggregation and distribution among the macro-queues and sub-queues organized on the router interface must solve the congestion management problem. Secondly, the resource allocation problem solution was related to the balanced allocation of interface bandwidth among the sub-queues, which were weighted relative to their priorities under the traffic-engineering queues. The method's lower-level functions were recommended to be placed on a set of processors of a routing device responsible for servicing the packets of individual macro-queues. At the same time, the processor coordinator could perform the functions of the upper-level calculations, providing interface bandwidth allocation among the macro-queues. The numerical research results of the proposed two-level hierarchical queue management method confirmed its effectiveness in ensuring high scalability. Balanced, priority-based packet flow distribution and interface bandwidth allocation among the macro-queues and sub-queues were implemented. In addition, the time was reduced for solving tasks related to queue management. The method demonstrated high convergence of the coordination procedure and the quality of the centralized calculations. The proposed approach can be used in various embedded systems.

Keywords: queue management; traffic priority; traffic engineering; optimization; network



Citation: Lemeshko, O.; Yeremenko, O.; Titarenko, L.; Barkalov, A. Hierarchical Queue Management Priority and Balancing Based Method under the Interaction Prediction Principle. *Electronics* **2023**, *12*, 675. <https://doi.org/10.3390/electronics12030675>

Academic Editor: Cheng-Chi Lee

Received: 21 December 2022

Revised: 23 January 2023

Accepted: 27 January 2023

Published: 29 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern communication networks are built as multilevel multiservice platforms, and their main task is still ensuring a given quality of service (QoS) for end users [1–4]. With growth in the territorial distribution of network devices (switches, routers, network controllers, etc.) in addition to increases in the volumes of network load and traffic heterogeneity, the problem of QoS provision only worsens. Each packet flow generated by a particular network application requires differentiated service and is specifically sensitive to specific QoS indicators [5,6]. For example, data traffic is traditionally critically sensitive to packet loss, and multimedia traffic is primarily sensitive to packet delays and jitter (delay variation). Nevertheless, any network traffic type requires a certain amount of bandwidth. Therefore, the primary architectural model for providing QoS in IP and MPLS networks is DiffServ, based on priority packet processing on routers [5–7].

As analyses have shown, the main technological means of ensuring the differentiated quality of service are congestion management mechanisms, which usually include FIFO, PQ, CQ, FQ/WFQ, CBQ, LLQ, and their numerical modifications and combinations [1–4,8–13]. At the moment, the perfect mechanism does not exist. Each has its

advantages, disadvantages, and recommended scope of application for various interfaces of switches and routers.

The traditional QoS approaches allocate resources based on service and traffic types. Indeed, the conventional QoS design, DiffServ, distributes packets into several queues based on how closely their priorities correspond to a device's priority [5–7]. Under this scheduling method, different queues are selected depending on the order in which packets are forwarded to the network device [14–16]. Due to the issues that classical QoS deals with related to the overgrown number of users, services, and network devices, hierarchical QoS (H-QoS) was introduced to address the existing limitations and provide QoS regarding various demands. Accordingly, H-QoS uses hierarchical scheduling, congestion management, and resource allocation under different traffic types, user classes, and priorities.

Analyses of existing works regarding hierarchical QoS have demonstrated significant interest in this class of solutions [1–4]. In addition, many developments are related to implementing efficient queue management strategies on network devices, namely, software-defined network programmable switches [1,10–13]. Particular attention should be paid to solutions related to priority-based queuing mechanisms [12,13] and load balancing under queue management [11,17,18].

Consequently, in the current work, we propose an improved two-level hierarchical queue management method based on priority and balancing under the interaction prediction principle while solving congestion management and resource allocation tasks. The main idea of the work is to provide an advanced method of hierarchical queue management to increase router performance through multiprocessor architectures. Since routers are embedded systems dedicated to forwarding packets efficiently from one network interface to another, the proposed approach can be used in various embedded systems of this type [14,16].

An advanced two-level hierarchical queue management method generally aims at increased scalability. While applying a multicore, multiprocessor architectures help improve overall performance by moving away from centralized and unreliable nonscalable solutions.

The remainder of this work is structured as follows: Section 2 defines a mathematical model of queue management with load balancing on routing devices in a communication network. Section 3 proposes a two-level hierarchical queue management method based on priority and balancing. Section 4 contains the numerical research of the proposed hierarchical queue management method under investigation with different sizes of macro-queue and sub-queue organization. Finally, Section 5 discusses the obtained research results regarding the coordination procedure iteration numbers of the proposed method, and Section 6 presents the conclusions of the work.

2. Mathematical Model of Queue Management with Load Balancing on Routing Devices

Within the proposed method, in addition to using the models of [17,18], the following consequent tasks must be solved:

- congestion management;
- resource allocation.

Suppose that, at the first calculation stage, M packet flows arrive at a router interface input with a known i th flow average intensity of a_i ($i = \overline{1, M}$) measured in bits per second. Then, a priority value of k_i^f ($i = \overline{1, M}$) corresponds to each i th packet flow. Assume that the flow priority is quantified by a number that varies from 0 to $K - 1$, where K is the maximum flow priority value.

For example, if packet processing and queue maintenance is based on DSCP (differentiated services code point) policies (Table 1), then $K = 64$. In the case of QoS group support, one hundred priorities $K = 100$ can be used on the router [19–22]. The higher the flow priority value k_i^f , the higher the QoS level that must be served on the interface.

Table 1. Correspondence between numeric values and names of DSCP policies.

DSCP Policy	Binary Value	Decimal Value	Standard
CS0	000000	0	RFC2474
CS1	001000	8	RFC2474
CS2	010000	16	RFC2474
CS3	011000	24	RFC2474
CS4	100000	32	RFC2474
CS5	101000	40	RFC2474
CS6	110000	48	RFC2474
CS7	111000	56	RFC2474
AF11	001010	10	RFC2597
AF12	001100	12	RFC2597
AF13	001110	14	RFC2597
AF21	010010	18	RFC2597
AF22	010100	20	RFC2597
AF23	010110	22	RFC2597
AF31	011010	26	RFC2597
AF32	011100	28	RFC2597
AF33	011110	30	RFC2597
AF41	100010	34	RFC2597
AF42	100100	36	RFC2597
AF43	100110	38	RFC2597
VOICE-ADMIT	101100	44	RFC5865
EF	101110	46	RFC3246

Let us introduce a two-level hierarchy of queues created and configured on a specific router interface with a bandwidth of B (bits per second). Let L macro-queues be organized on the interface. Every macro-queue is divided into sub-queues N_l ($l = \overline{1, L}$) according to the established traffic classification system and the supported level of QoS differentiation. Then, the total number of sub-queues on the interface is $N = \sum_{l=1}^L N_l$.

Priority-based queuing is grounded on the involvement of the queue priority concept, which should be directly related to the packet flow priority. Then, we introduce the following parameters for each of the sub-queues of any macro-queue:

- $K_{j,l}^{\min}$ and $K_{j,l}^{\max}$ are the minimum and maximum values of the packet flow priority that the l th macro-queue's j th sub-queue can serve, respectively;
- $K_{j,l}$ is the total number of packet flow priorities that the l th macro-queue's j th sub-queue can serve ($l = \overline{1, L}$).

The parameters of $K_{j,l}$, $K_{j,l}^{\min}$, and $K_{j,l}^{\max}$ are positive integers related by the following equation:

$$K_{j,l} = K_{j,l}^{\max} - K_{j,l}^{\min} + 1 \quad (j = \overline{1, N_l}, l = \overline{1, L}).$$

The ranges of the priority values $K_{j,l}^{\min}$ and $K_{j,l}^{\max}$ between sub-queues and macro-queues can be distributed statically or dynamically according to various criteria, for example, evenly. When arriving at the interface of a packet flow that has a k_i^f priority, it is immediately directed to the l th macro-queue's j th sub-queue, for which the condition $K_{j,l}^{\min} \leq k_i^f \leq K_{j,l}^{\max}$ is fulfilled. In fact, the $k_{j,l}^q$ priority of the j th sub-queue of the l th macro-queue is the arithmetic mean of $K_{j,l}^{\min}$ and $K_{j,l}^{\max}$.

Thus, a set of packet flows is formed, which is sent to one or another macro-queue (Table 2). M_l denotes the total number of packet flows sent to the l th macro-queue due to distribution and aggregation. Flows are aggregated by the sub-queues of one macro-queue if $M_l > N_l$.

Table 2. Queue prioritization and congestion management problem-solving algorithm.

Queue prioritization and Congestion Management	
1:	Inputs: L, N_l, K, M
2:	for $l = 1, 2, \dots, L$ calculate % macro-queue number
3:	for $j = 1, 2, \dots, N_l$ calculate % sub-queue number
4:	Determine $K_{j,l}^{\min}, K_{j,l}^{\max}$, and $k_{j,l}^q$
5:	end for
6:	end for
7:	for $i = 1, 2, \dots, M$ calculate % packet flow number
8:	for $l = 1, 2, \dots, L$ calculate % macro-queue number
9:	for $j = 1, 2, \dots, N_l$ calculate % sub-queue number
10:	if $K_{j,l}^{\min} \leq k_i^f \leq K_{j,l}^{\max}$
11:	$x_i^{j,l} = 1$
12:	else $x_i^{j,l} = 0$
13:	end if
14:	end for
15:	end for
16:	end for
17:	Outputs: $K_{j,l}^{\min}, K_{j,l}^{\max}, k_{j,l}^q, x_i^{j,l}$, and M_l

The result of the algorithm application (Table 2) determines the solution to the congestion management problem by defining a set of variables $x_i^{j,l} \in \{0, 1\}$ ($i = \overline{1, M}, j = \overline{1, N_l}, l = \overline{1, L}$), each of which characterizes the fraction of the i th flow sent for servicing to the j th sub-queue of the l th macro-queue [17,18]. In most queue-scheduling mechanisms, such as PQ, CQ, CBQ, and LLQ, the administrator solves the congestion management problem by setting, for example, ACL (access control lists) [6].

After solving the problem of the optimal aggregation and distribution of packet flows among the macro-queues and sub-queues represented by a set of calculated values $x_i^{j,l}$, resource allocation is performed, which relates to the second stage of calculations. Next, we have to introduce the following control variables to solve the resource allocation problem:

- b_l ($l = \overline{1, L}$) defines the interface bandwidth allocated for servicing the l th macro-queue;
- $b_{j,l}$ ($j = \overline{1, N_l}, l = \overline{1, L}$) defines the interface bandwidth allocated for servicing the l th macro-queue's j th sub-queue.

Following their physical sense, the variables b_l and $b_{j,l}$ are subject to the following constraints, respectively:

$$0 \leq b_l, \sum_{l=1}^L b_l = B, \tag{1}$$

$$0 \leq b_{j,l}, \sum_{j=1}^{N_l} b_{j,l} = b_l \quad (l = \overline{1, L}). \tag{2}$$

Compliance with conditions (1) and (2) indicates proper bandwidth interface allocation among the macro-queues and sub-queues.

Additionally, it is necessary to satisfy the nonlinear conditions of sub-queue overload prevention by the bandwidth allocated to them to ensure optimal allocation and interface bandwidth balancing among the sub-queues under the traffic-engineering queue concept [17,18]:

$$h_{j,l}^\alpha \sum_{i=1}^M a_i x_i^{j,l} \leq \alpha_l b_{j,l} \quad (j = \overline{1, N_l}, l = \overline{1, L}), \tag{3}$$

where α_l is the control variable quantified with the upper dynamically controlled bound of the l th macro-queue's sub-queues utilization by bandwidth under the following condition:

$$0 < \alpha_l \leq 1. \tag{4}$$

In turn, $h_{j,l}^\alpha$ is the priority coefficient introduced to ensure balanced interface bandwidth allocation among the l th macro-queue sub-queues considering their priorities:

$$h_{j,l}^\alpha = 1 + \frac{k_{j,l}^q}{K \cdot D} \quad (j = \overline{1, N_l}, l = \overline{1, L}), \tag{5}$$

where $D > 0$ is the normalization coefficient, which determines the level of influence of the queue priority $k_{j,l}^q$ on the priority coefficient $h_{j,l}^\alpha$. and the process of bandwidth balancing among the sub-queues.

The higher the queue priority $k_{j,l}^q$, the higher the value of $h_{j,l}^\alpha$. Thus, the higher the queue priority $h_{j,l}^\alpha$, the smaller its utilization $\rho_{j,l}$ for the same boundary value α_l . In model notations (1)–(5), the utilization coefficient is determined by the following formula [17,18]:

$$\rho_{j,l} = \frac{\sum_{i=1}^M a_i x_i^{j,l}}{b_{j,l}} \quad (j = \overline{1, N_l}, l = \overline{1, L}). \tag{6}$$

The higher the normalization coefficient D , the less queue priority affects the allocated bandwidth volume. Based on the introduction of expressions (3)–(6), the differentiation in the bandwidth allocation of the router interface among the priority sub-queues organized on it is provided.

In turn, the nonlinearity of condition (3) is determined by the presence in the right-hand side of a bilinear form—the product of control variables $b_{j,l}$ and α_l . In this case, all the parameters on the left side of condition (3) are known values. The threshold α_l allows balancing the bandwidth required for service. Condition (3) demonstrates the functional relations of control variables during calculation.

Then, constraint (3) to move to a linear form can be represented as follows:

$$\alpha_l^* h_{j,l}^\alpha \sum_{i=1}^N a_i x_i^{j,l} \leq b_{j,l} \quad (j = \overline{1, N_l}, l = \overline{1, L}), \tag{7}$$

where α^* is an additional control variable introduced that is inversely proportional to the upper bound of the interface queue utilization (α), i.e.:

$$\alpha_l^* = \frac{1}{\alpha_l}. \tag{8}$$

The following restrictions are imposed on this variable:

$$\alpha_l^* > 0. \tag{9}$$

Accordingly, based on the known order of flow aggregation and distribution defined by the variables $x_i^{j,l}$, it is necessary to determine the order of interface bandwidth distribution among the macro-queues and sub-queues following conditions (1)–(9).

3. Two-Level Hierarchical Queue Management Method Based on Priority and Balancing

To solve the resource allocation problem, which is primarily related to the calculation of the set of variables of b_l and $b_{j,l}$ ($j = \overline{1, N_l}, l = \overline{1, L}$), the interaction prediction principle, which is part of the hierarchical multilevel control systems theory, was used in this work [23]. The interaction prediction principle, which involves a multilevel calculations hierarchy,

aims to increase queue management solutions’ scalability when the separate processors (cores) of a router’s computing system perform macro-queue management tasks.

Hence, a two-level decision hierarchy was introduced for models (1)–(9). According to the interaction prediction principle at the top hierarchical level, the problem of calculating the interface’s bandwidth allocated to macro-queues ($b_l, j = \overline{1, L}$) was solved. The lower level was responsible for the bandwidth b_l distribution of macro-queues obtained from the upper-level calculations among the corresponding sub-queues by defining variables $b_{j,l}$ ($j = \overline{1, N_l}, l = \overline{1, L}$).

The proposed hierarchical queue management method based on priority and balancing (Figure 1) was established on the following iterative sequence of actions.

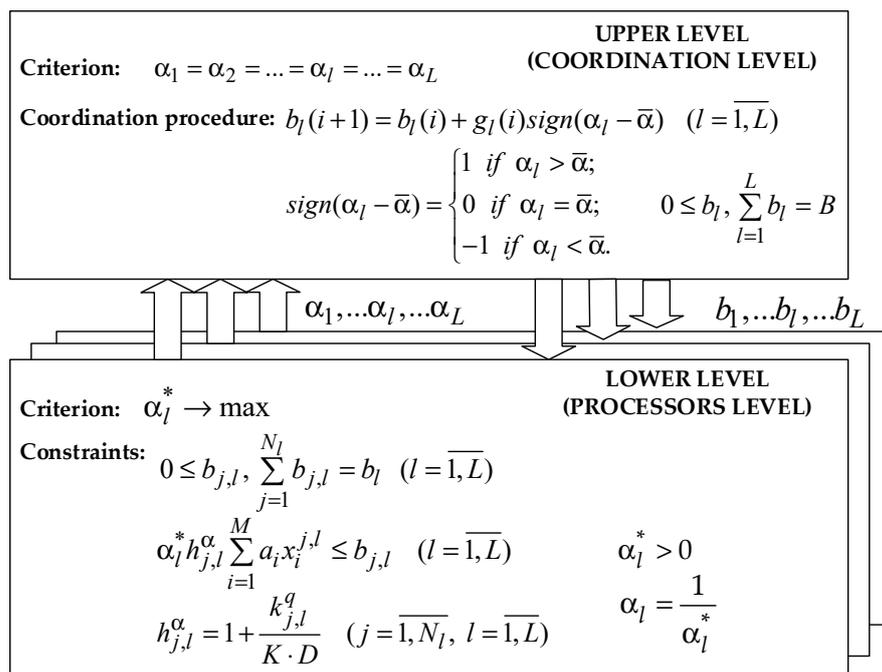


Figure 1. The scheme of the two-level hierarchical queue management method based on priority and balancing.

At the *zero stage* of the method, the initial conditions for solving the resource allocation problem were set: at the top level of calculations, the interface bandwidth was allocated for each macro-queue (b_l) in such a way that condition (1) was fulfilled. Allocation of the router interface bandwidth among the macro-queues at this iteration could be performed uniformly or proportionally to the volume or priority of the load arriving at the macro-queues.

At the *first stage* of the method, for the lower-level calculations, namely, those at the level of the individual processors (cores) of a router’s computing system, the variables $b_{j,l}$ ($j = \overline{1, N_l}, l = \overline{1, L}$) were simultaneously determined for every macro-queue by solving a linear programming optimization problem, where the optimality criterion was the maximum variable α_l^* introduced in (7):

$$\alpha_l^* \rightarrow \max \tag{10}$$

resulting in the fulfillment of constraints (2), (7), and (9) when $x_i^{j,l}$ are known values (Figure 1). Taking into account (8) and (10), at this level the upper bound of the sub-queue utilization was minimized for each macro-queue ($\alpha_l \rightarrow \min$).

The satisfaction of requirements (2), (7), and (9) when minimizing the bound α_l (4) under maximizing the variable α_l^* (10) allowed for providing an optimum balanced router interface bandwidth distribution among the l th macro-queue sub-queues formed under the principles of the traffic-engineering queues [17,18]. Therefore, at the lower level, the

variables of α_l and $b_{j,l}$ ($j = \overline{1, N_l}, l = \overline{1, L}$) were calculated and, at the same time, α_l were transferred to the upper hierarchical level for the subsequent coordination of the obtained solutions by updating the values of b_l ($l = \overline{1, L}$).

At the *second stage* of the method, the variables of b_l , ($l = \overline{1, L}$) were adjusted in order to achieve a quality level of centralized calculations so that the following condition was met:

$$\alpha_1 = \alpha_2 = \dots = \alpha_l = \dots = \alpha_L. \tag{11}$$

The conditions sensed that the values of the utilization upper bounds of different macro-queue sub-queues, which were weighted relative to their priority (3), should be the same. Consequently, it was proposed to modify the variables b_l , ($l = \overline{1, L}$) based on the use of the following iterative search procedure:

$$b_l(i + 1) = b_l(i) + g_l(i) \text{sign}(\alpha_l - \bar{\alpha}) \quad (l = \overline{1, L}), \tag{12}$$

where i is the search iteration number; $g_l(i)$ is the search step length selected according to the search procedure convergence conditions (9); and $\bar{\alpha}$ is the average value of the utilization bounds of the macro-queue sub-queues (4):

$$\text{sign}(\alpha_l - \bar{\alpha}) = \begin{cases} 1 & \text{if } \alpha_l > \bar{\alpha}; \\ 0 & \text{if } \alpha_l = \bar{\alpha}; \\ -1 & \text{if } \alpha_l < \bar{\alpha}. \end{cases} \tag{13}$$

Thus, the higher the utilization upper bound of the sub-queues of a specific macro-queue, the more interface bandwidth allocated to this macro-queue (12). Conversely, if the macro-queue utilization bound is lower than the average value $\bar{\alpha}$, the interface bandwidth allocated to it decreases.

The updated variable b_l ($l = \overline{1, L}$) values descended to the lower level of calculations to obtain the new values of α_l and $b_{j,l}$ ($j = \overline{1, N_l}, l = \overline{1, L}$). That is, the method operation took on an iterative nature. The completion of the method occurred when condition (11) was met at the upper level of calculations. In this case, the function (13) value for any macro-queue was zero.

4. Numerical Research

A study of the two-level queue management priority-based traffic-engineering method presented in the previous section was conducted to evaluate the effectiveness of the solutions obtained and to analyze the convergence speed of coordination procedures (12) and (13).

For clarity, organizations of different numbers of macro-queues (from two to six) on the interface, sub-queues, and packet flows are considered and investigated. We dwell on the analyses of cases with the organization of three and five macro-queues.

4.1. Organization of Three Macro-Queues

We organized three macro queues $L = 3$ on an interface ($B = 100$ Mbps). Each macro-queue was divided into three sub-queues of $N_1 = N_2 = N_3 = 3$. Fifteen packet flows with the intensities (Mbps) and DSCP priorities given in Table 3 were sent to this interface following the contents of the routing table.

Table 3. Packet flow parameters.

Flow #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
α_i	6.4	6	7.2	6.4	8	4.4	4.4	6	6.4	5.6	3.2	4.8	3.6	3.6	4
k_i^f	1	6	12	17	20	26	31	35	40	41	47	50	53	58	61

The normalization coefficient D equal to 8, as well as the flow priorities ($K_{j,l}^{\min}, K_{j,l}^{\max}$) and sub-queues ($k_{j,l}^q$), which were distributed evenly among them, are presented in Table 4. Since the flow priorities (Table 1) were whole numbers, seven flow priorities were assigned to each sub-queue, and eight packet flow priorities were set to the last (highest priority) one.

Table 4. Three macro-queues’ sub-queue priorities.

Macro Queue #	1			2			3		
Sub-queue #	1	2	3	1	2	3	1	2	3
Flow priorities range	[0, 6]	[7, 13]	[14, 20]	[21, 27]	[28, 34]	[35, 41]	[42, 48]	[49, 55]	[56, 63]
Sub-queue priority	3	10	17	24	31	38	45	52	59

According to the selected algorithm (Table 2), the order of the 15 flows’ aggregation and distribution among the sub-queues of macro-queues presented in Table 5 was obtained. In Table 5, the belonging of a packet flow to one or another sub-queue of a macro-queue is marked in gray color. For example, the first and second flows were directed to the first macro-queue’s first sub-queue, etc.

Table 5. The 15 flows’ aggregation and distribution among the sub-queues of three macro-queues.

Macro-queue 1	Flow #				
	1	2	3	4	5
1					
2					
3					
Macro-queue 2	Flow #				
	6	7	8	9	10
1					
2					
3					
Macro-queue 3	Flow #				
	11	12	13	14	15
1					
2					
3					

During the study of the proposed two-level method, coordinated solutions (Table 6) were obtained for certain numbers of iterations of coordination procedures (12) and (13), which were influenced by the degree of closeness of the signature value (13) to zero. Table 6 shows the results of the method for each of the four iterations when the difference (δ) in the function (11) argument values was less than 0.0001 of the absolute value.

Table 6. Method application results for four coordination iterations.

Iteration #	α_1	α_2	α_3	$\bar{\alpha}$	b_1	b_2	b_3
1	0.8163	0.8540	0.8838	0.8513	42.5000	33.5000	24.0000
2	0.8441	0.8513	0.8385	0.8446	41.0972	33.6060	25.2968
3	0.8445	0.8446	0.8467	0.8453	41.0774	33.8727	25.0499
4	0.8451	0.8451	0.8451	0.8451	41.0476	33.8539	25.0984

For the last (fourth) iteration of calculations, Table 7 shows the order of interface bandwidth allocation among the sub-queues of three macro-queues and their utilization (6).

Table 7. The interface bandwidth allocation among the sub-queues of three macro-queues and their utilization.

Macro-Queue #	1			2			3		
Sub-queue #	1	2	3	1	2	3	1	2	3
Priority	3	10	17	24	31	38	45	52	59
Aggregated flow intensity	12.4	7.2	14.4	4.4	4.4	18.0	3.2	8.4	7.6
Bandwidth	14.7579	8.6856	17.6041	5.4508	5.5220	22.8812	4.1194	10.9494	10.0296
Utilization	0.8402	0.8290	0.8180	0.8072	0.7968	0.7867	0.7768	0.7672	0.7578

As can be seen from Figure 2 and Table 7, queues were loaded in a balanced manner while taking into account their priority level. A higher-priority queue always had lower utilization (6) than a lower-priority queue. In Figure 2, for example, the queue number “2|1” indicates that this was the first sub-queue of the second macro-queue.

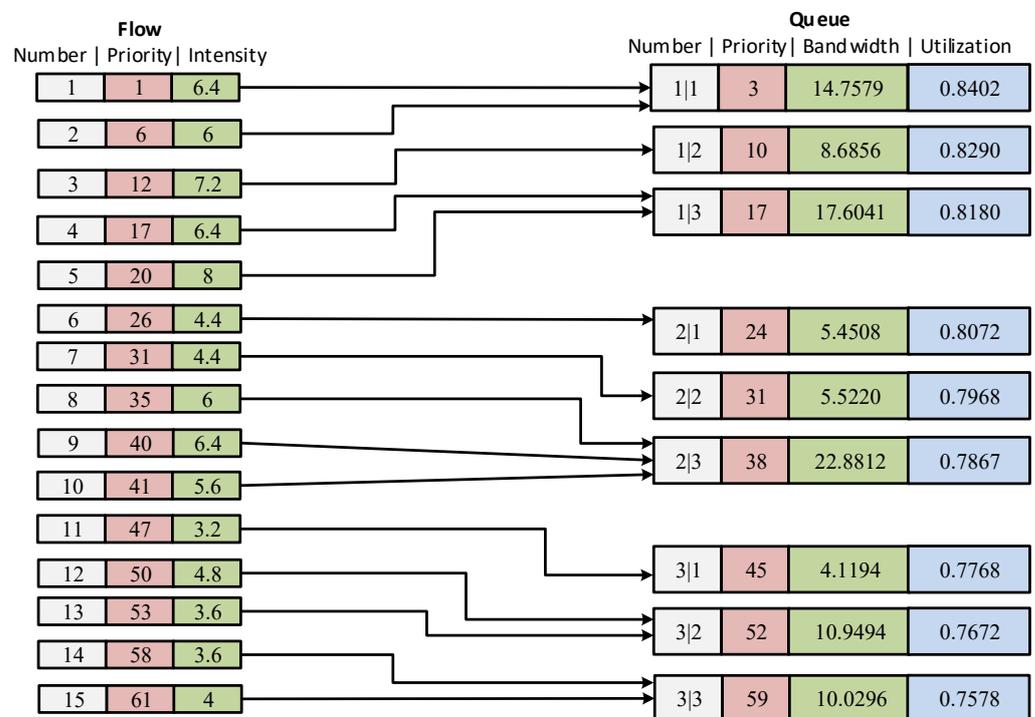


Figure 2. The resulting solution for interface bandwidth allocation among the sub-queues of three macro-queues.

It was established experimentally that the minimum value of parameter D (5) at which the adequacy of models (1)–(9) was ensured and condition (3) was fulfilled depended firstly on interface utilization and secondly on the number of macro-queues (Table 8). At the same time, the interface utilization was calculated as $\left(\sum_{i=1}^M a_i\right) / B$.

For this example, the choice of the quantitative value of parameter D was justified by the need to ensure high differentiation in packets serving in different queues. Figure 3 shows the dependence of queue utilization for nine priority sub-queues on normalization coefficient D . At the minimum values of D , the maximum differentiation in the services of distinct sub-queues was ensured. As D increased, the difference in the utilization of each queue was minimized.

Table 8. Dependence of the minimum value D on the interface utilization and number of macro-queues organized on the interface.

Interface Utilization	Number of Macro-Queues on the Interface				
	2	3	4	5	6
	Minimum Value of D				
0.6	1.1	1.3	1.4	1.4	1.7
0.7	1.6	2	2.1	2.1	2.3
0.8	2.6	3.4	3.5	3.6	3.9
0.9	5.1	7.6	7.9	8	8.5
0.95	10.7	16	16.7	16.8	18

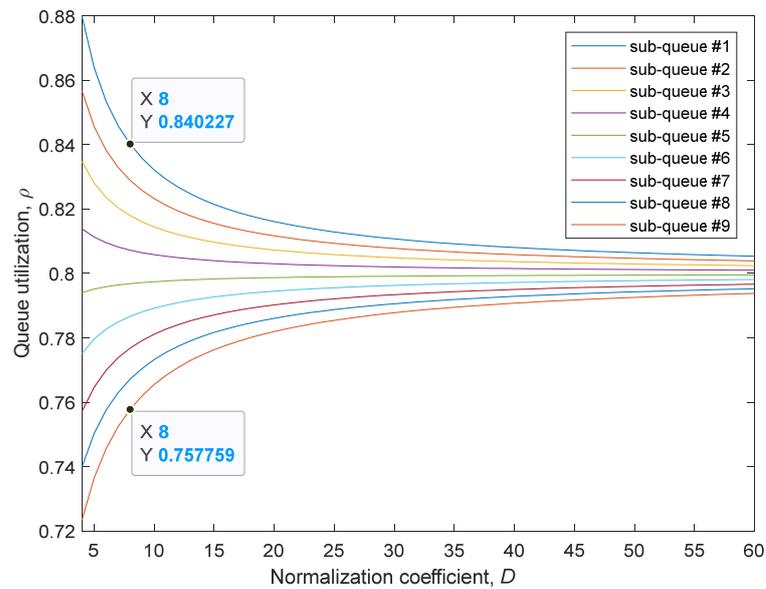


Figure 3. Dependence of queue utilization for nine priority sub-queues on normalization coefficient D .

4.2. Organization of Five Macro-Queues

We also demonstrated the features of the proposed method when five macro-queues ($L = 5$) were organized on an interface ($B = 100$ Mbit/s), each divided into five sub-queues ($N_1 = N_2 = N_3 = N_4 = N_5 = 5$). From the content of the routing table, forty packet flows were sent to this interface. The flow intensities (Mbps) and DSCP priorities are shown in Table 9.

Table 9. Packet flow parameters.

Flow #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
α_i	2	3	1	2	4	2	1	3	2	4	1	1	2	1
k_i^f	1	3	4	5	6	7	8	9	11	12	13	14	15	17
Flow #	15	16	17	18	19	20	21	22	23	24	25	26	27	-
α_i	1	3	3	1	2	1	3	2	1	1	2	1	2	-
k_i^f	18	19	21	22	25	27	28	30	31	32	35	36	38	-
Flow #	28	29	30	31	32	33	34	35	36	37	38	39	40	-
α_i	1	1	2	2	3	3	1	3	4	1	1	2	4	-
k_i^f	41	42	44	46	47	50	52	54	56	57	59	61	62	-

The normalization coefficient D was equal to 10, with priorities of flows ($K_{j,l}^{\min}, K_{j,l}^{\max}$) and sub-queues ($k_{j,l}^q$) distributed evenly between them, as presented in Table 10.

Table 10. Three macro-queues’ sub-queue priorities.

Macro-queue #	1					2				
Sub-queue #	1	2	3	4	5	1	2	3	4	5
Flow priorities range	[0, 1]	[2, 3]	[4, 5]	[6, 7]	[8, 9]	[10, 11]	[12, 13]	[14, 15]	[16, 17]	[18, 19]
Sub-queue priority	0.5	2.5	4.5	6.5	8.5	10.5	12.5	14.5	16.5	18.5
Macro-queue #	3					4				
Sub-queue #	1	2	3	4	5	1	2	3	4	5
Flow priorities range	[20, 21]	[22, 24]	[25, 27]	[28, 30]	[31, 33]	[34, 36]	[37, 39]	[40, 42]	[43, 45]	[46, 48]
Sub-queue priority	20.5	23	26	29	32	35	38	41	44	47
Macro-queue #	5									
Sub-queue #	1	2	3	4	5					
Flow priorities range	[49, 51]	[52, 54]	[55, 57]	[58, 60]	[61, 63]					
Sub-queue priority	50	53	56	59	62					

According to the selected algorithm (Table 2), the order of the 40 flows’ aggregation and distribution among the sub-queues of macro queues presented in Table 11 was obtained. Similar to Table 5, in Table 11, the belonging of a packet flow to one or another sub-queue of a macro-queue is marked in gray color. The first flow was directed to the first macro-queue’s first sub-queue, the second flow was directed to the first macro-queue’s second sub-queue, etc.

Table 11. The 40 flows’ aggregation and distribution among the sub-queues of five macro-queues.

Macro-queue 1	Flow #							
	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
Macro-queue 2	Flow #							
	9	10	11	12	13	14	15	16
1								
2								
3								
4								
5								
Macro-queue 3	Flow #							
	17	18	19	20	21	22	23	24
1								
2								
3								
4								
5								
Macro-queue 4	Flow #							
	25	26	27	28	29	30	31	32
1								
2								
3								
4								
5								
Macro-queue 5	Flow #							
	33	34	35	36	37	38	39	40
1								
2								
3								
4								
5								

Table 12 shows the results of the proposed method’s application for each iteration when the differences (δ) in the values of function (11) arguments were also less than 0.0001 by absolute value.

Table 12. Method application results for five coordination iterations.

Iteration #	α_1	α_2	α_3	α_4	α_5	$\bar{\alpha}$
1	0.8066	0.8181	0.8332	0.8523	0.8706	0.8362
2	0.8312	0.8450	0.8379	0.8270	0.8413	0.8365
3	0.8358	0.8362	0.8364	0.8369	0.8376	0.8366
4	0.8365	0.8365	0.8365	0.8367	0.8369	0.8366
5	0.8366	0.8366	0.8366	0.8366	0.8366	0.8366
Iteration #	b_1	b_2	b_3	b_4	b_5	-
1	22.5	18.75	17.5	17.5	23.75	-
2	21.8346	18.153	17.401	18.0349	24.5765	-
3	21.7149	18.3451	17.4328	17.8221	24.685	-
4	21.6967	18.3392	17.4299	17.8274	24.7068	-
5	21.6937	18.337	17.4288	17.8285	24.7121	-

For the last (fifth) iteration of calculations, Table 13 shows the order of interface bandwidth allocation among the sub-queues of five macro-queues and their utilization (6).

Table 13. The interface bandwidth allocation among the sub-queues of five macro-queues and their utilization.

Macro-queue #	1					2				
Sub-queue #	1	2	3	4	5	1	2	3	4	5
Priority	0.5	2.5	4.5	6.5	8.5	10.5	12.5	14.5	16.5	18.5
Aggregated flow intensity	2	3	3	6	4	2	5	3	1	4
Bandwidth	2.3926	3.6000	3.6113	7.2449	4.8449	2.4300	6.0936	3.6674	1.2262	4.9197
Utilization	0.8359	0.8333	0.8307	0.8282	0.8256	0.8230	0.8205	0.8180	0.8155	0.8131
Macro-queue #	3					4				
Sub-queue #	1	2	3	4	5	1	2	3	4	5
Priority	20.5	23	26	29	32	35	38	41	44	47
Aggregated flow intensity	3	1	3	5	2	3	2	2	2	5
Bandwidth	3.7009	1.2383	3.7317	6.2476	2.5102	3.7820	2.5325	2.5437	2.5549	6.4154
Utilization	0.8106	0.8076	0.8039	0.8003	0.7967	0.7932	0.7897	0.7862	0.7828	0.7794
Macro-queue #	5									
Sub-queue #	1	2	3	4	5					
Priority	50	53	56	59	62					
Aggregated flow intensity	3	4	5	1	6					
Bandwidth	3.8656	5.1766	6.4988	1.3054	7.8657					
Utilization	0.7761	0.7727	0.7694	0.7661	0.7628					

The obtained results (Table 13) also confirmed the effectiveness of the proposed method in ensuring balanced queue loading, taking into account their priority level. For example, the fifth sub-queue of the fifth macro-queue had the highest priority of 62 and the lowest utilization of 0.7628. At the same time, the first sub-queue of the first macro-queue had the lowest priority of 0.5 and the highest utilization of 0.8359.

5. Discussion

Therefore, the proposed method demonstrated reasonably fast convergence. The solutions presented in Tables 7 and 13 fully corresponded to the level of the centralized calculations. Thus, the decentralization of calculations among the separate processors

(cores) of a router’s computing system did not affect the quality of the obtained solutions. Such a result confirmed the advantage of using the interaction prediction principle when coordinating solutions obtained at different hierarchical levels of the method (Figure 1).

At the same time, the link resource between the macro-queues and sub-queues was distributed and balanced under their priorities, that is, the higher the sub-queue priority, the lower its utilization (Tables 7 and 13), which directly affected the quality of service level of packets in this queue.

Figure 4 presents the dependence of coordination procedures (12) and (13) on iteration number, which were required for method convergence to the optimal solution, with the number of macro-queues at $\delta < 0.0001$.

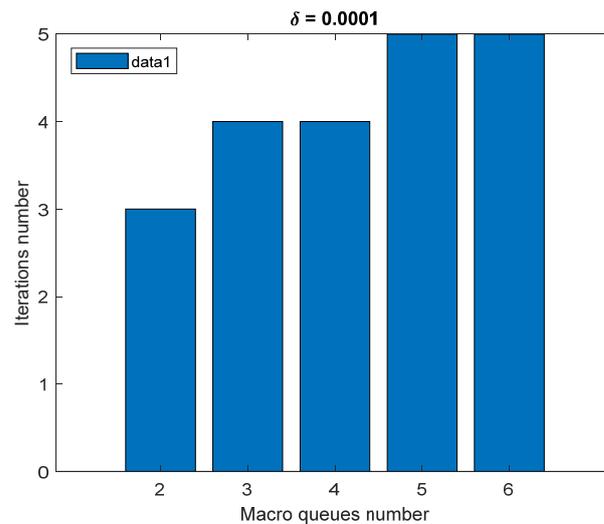


Figure 4. Dependence of coordination procedures (12) and (13) on iteration number required for method convergence to the optimal solution with the number of macro-queues at $\delta < 0.0001$.

With a decrease in the accuracy requirements of condition (11), the iteration number slightly decreased by an average of 20% to 30% (Figure 5). However, this did not significantly change the nature of the bandwidth allocation among the macro-queues or among the individual sub-queues: the difference in the final decisions ranged from 0.55% to 1.1%. Thus, the proposed method demonstrated reasonably fast convergence within the proposed numerical examples from one to five iterations.

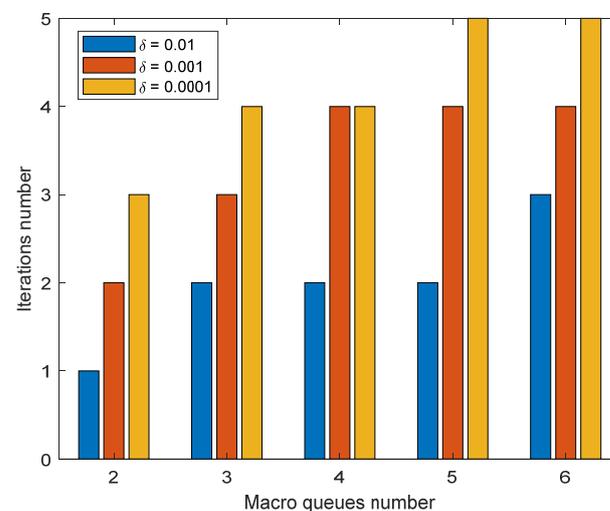


Figure 5. Dependence of coordination procedures (12) and (13) on iteration number required for method convergence to the optimal solution with the number of macro-queues.

A positive feature of the interaction prediction principle used in the method was that any solution obtained at an intermediate iteration could be physically implemented. It may not be optimal, but its implementation would lead to link resource overload.

6. Conclusions

Hierarchical queues are increasingly utilized to improve the scalability of solutions regarding queue management on router interfaces. On the other hand, to increase router performance, which has to serve gigabit and sometimes terabit flows in real time, these devices are often built on multiprocessor (multicore) architecture. Therefore, decisions regarding queue management must consider the possibility of distributed (parallel) computing, which can also be effectively implemented based on hierarchical queues.

In consequence, this work proposed a two-level hierarchical queue management method based on priority and balancing. The method was grounded on the interaction prediction principle for coordinating different levels of decisions. The lower level of calculations, which was based on the problem optimization of solutions (1)–(7), was responsible firstly for the aggregation and distribution of packet flows among the macro-queues and sub-queues organized on the router interface (congestion management problem) and secondly for the balanced allocation of interface bandwidth among sub-queues, which were weighted relative to their priorities (resource allocation problem). The problem of balanced router interface bandwidth allocation among the priority sub-queues was solved by considering the requirements of traffic-engineering queues. It was advisable to place the lower-level functions of the method on a set of processors (cores), which were responsible for servicing the packets of individual macro-queues. The upper level of the method's calculations was responsible for interface bandwidth allocation among the macro-queues by performing the iterative procedures of (12) and (13). The processor coordinator could perform the functions of the upper-level calculations.

The numerical research results of the proposed two-level hierarchical queue management method based on priority and balancing confirmed its effectiveness in ensuring high scalability, balanced and priority-based distribution of packet flows, and interface bandwidth allocation among the macro-queues and sub-queues organized on routers. The method provided a functional decomposition of low-level computational tasks among the processors (cores) of a router, allowing them to be solved simultaneously and improving the time for solving tasks related to queue management. Within the considered example, the method demonstrated high convergence of coordination procedures (12) and (13), which were carried out for 1–5 iterations (Figure 5), and the final quality of the centralized calculations.

Our future research is concerned with improving the presented method by enhancing its flexibility and moving on to three-level solutions considering multiprocessing and multicore problems. In addition, possible modifications can be connected with the updated mathematical model using other types of coordination. The practical application of the proposed approach is mainly related to programmable networks where vast amounts of user data flow must be served efficiently [24,25]. At the same time, technological solutions must satisfy the demands for scalability and quality of service.

Author Contributions: Conceptualization, O.L., O.Y., L.T. and A.B.; software, O.L. and O.Y.; validation, O.L. and A.B.; formal analysis, O.L., O.Y., L.T. and A.B.; investigation, O.L. and O.Y.; resources, O.L. and O.Y.; data curation, L.T. and A.B.; writing—original draft preparation, O.L., O.Y., L.T. and A.B.; writing—review and editing, O.L., O.Y., L.T. and A.B.; visualization, O.L., O.Y., L.T. and A.B.; supervision, O.L., O.Y. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACL	Access Control Lists
CBQ	Class-Based Queuing
CQ	Custom Queueing
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point
FIFO	First In, First Out
FQ	Fair Queuing
H-QoS	Hierarchical Quality of Service
IP	Internet Protocol
LLQ	Low-Latency Queueing
MPLS	Multiprotocol Label Switching
PQ	Priority Queuing
QoS	Quality of Service
WFQ	Weighted Fair Queueing

References

- Fejes, F.; Nadas, S.; Gombos, G.; Laki, S. DeepQoS: Core-Stateless Hierarchical QoS in Programmable Switches. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 1842–1861. [\[CrossRef\]](#)
- Chowdhury, R.R.; Chattopadhyay, S.; Adak, C. CAHPHF: Context-Aware Hierarchical QoS Prediction with Hybrid Filtering. *IEEE Trans. Serv. Comput.* **2020**, *15*, 2232–2247. [\[CrossRef\]](#)
- Li, D.; Wang, W.; Kang, Y. A Hierarchical Approach for QoS-Aware Edge Service Scheduling and Composition. In Proceedings of the 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 27–29 August 2021; pp. 677–681. [\[CrossRef\]](#)
- You, C.; Zhao, Y.; Feng, G.; Quek, T.Q.S.; Li, L. Hierarchical Multi-resource Fair Queueing for Packet Processing. *IEEE Trans. Netw. Serv. Manag.* **2022**, 1–15. [\[CrossRef\]](#)
- Medhi, K.D. *Ramasamy, Network routing: Algorithms, Protocols, and Architectures*; Morgan Kaufmann: San Francisco, CA, USA, 2017.
- QoS: Congestion Management Configuration Guide, Cisco IOS XE Everest 16.5*; Cisco Systems, Inc.: San Jose, CA, USA, 2019.
- Park, G.; Jeon, B.; Lee, G.M. QoS Implementation with Triple-Metric-Based Active Queue Management for Military Networks. *Electronics* **2022**, *12*, 23. [\[CrossRef\]](#)
- Kattepur, A.; David, S.; Mohalik, S.K. Model-based reinforcement learning for router port queue configurations. *Intell. Conver. Netw.* **2021**, *2*, 177–197. [\[CrossRef\]](#)
- Zhang, Z.; Shi, P.; Ward, A.R. Routing for Fairness and Efficiency in a Queueing Model with Reentry and Continuous Customer Classes. In Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA, 8–10 June 2022. [\[CrossRef\]](#)
- Huang, Y.; Wang, S.; Zhang, X.; Huang, T.; Liu, Y. Flexible Cyclic Queueing and Forwarding for Time-Sensitive Software-Defined Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**. [\[CrossRef\]](#)
- Boero, L.; Cello, M.; Garibotto, C.; Marchese, M.; Mongelli, M. BeaqoS: Load balancing and deadline management of queues in an OpenFlow SDN switch. *Comput. Netw.* **2016**, *106*, 161–170. [\[CrossRef\]](#)
- Rahouti, M.; Xiong, K.; Xin, Y.; Ghani, N. A Priority-Based Queueing Mechanism in Software-Defined Networking Environments. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–2. [\[CrossRef\]](#)
- Singh, D.; Ng, B.; Lai, Y.-C.; Lin, Y.-D.; Seah, W.K. Modelling Software-Defined Networking: Switch Design with Finite Buffer and Priority Queueing. In Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, 9–12 October 2017; pp. 567–570. [\[CrossRef\]](#)
- Barkalov, A.; Titarenko, L.; Mazurkiewicz, M. *Foundations of Embedded Systems. Studies in Systems, Decision and Control*; Springer: Cham, Switzerland, 2019; Volume 195, p. 167. [\[CrossRef\]](#)
- Wang, J.; Lv, G.; Liu, Z.; Yang, X. Programmable Deterministic Zero-Copy DMA Mechanism for FPGA Accelerator. *Appl. Sci.* **2022**, *12*, 9581. [\[CrossRef\]](#)
- Adhi, B.; Cortes, C.; Tan, Y.; Kojima, T.; Podobas, A.; Sano, K. The Cost of Flexibility: Embedded versus Discrete Routers in CGRAs for HPC. In Proceedings of the 2022 IEEE International Conference on Cluster Computing (CLUSTER), Heidelberg, Germany, 6–9 September 2022; pp. 347–356. [\[CrossRef\]](#)
- Lemeshko, O.; Lebedenko, T.; Nevzorova, O.; Sniurov, A.; Mersni, A.; Al-Dulaimi, A. Development of the Balanced Queue Management Scheme with Optimal Aggregation of Flows and Bandwidth Allocation. In Proceedings of the 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, Ukraine, 26 February–2 March 2019; pp. 1–4. [\[CrossRef\]](#)

18. Lemeshko, O.; Lebedenko, T.; Mersni, A.; Hailan, A.M. Mathematical Optimization Model of Congestion Management, Resource Allocation and Congestion Avoidance on Network Routers. In Proceedings of the 2019 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), Odessa, Ukraine, 9–13 September 2019; pp. 1–5. [[CrossRef](#)]
19. Nichols, K.; Blake, S.; Baker, F.; Black, D. RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. 1998. Available online: <https://www.rfc-editor.org/rfc/rfc2474> (accessed on 2 September 2022).
20. Heinanen, J.; Baker, F.; Weiss, W.; Wroclawski, J. RFC2597: Assured Forwarding PHB Group. 1999. Available online: <https://www.rfc-editor.org/rfc/rfc2597> (accessed on 2 September 2022).
21. Baker, F.; Polk, J.; Dolly, M. A differentiated Services Code Point (dscp) for Capacity-Admitted Traffic (No. rfc5865). Available online: <https://www.rfc-editor.org/rfc/rfc5865.html> (accessed on 2 September 2022).
22. Davie, B.; Charny, A.; Bennet, J.C.R.; Benson, K.; Boudec, J.L.; Courtney, W.; Davari, S.; Firoiu, V.; Stiliadis, D. Rfc3246: An Expedited Forwarding PHB (Per-Hop Behavior). 2002. Available online: <https://www.rfc-editor.org/rfc/rfc3246> (accessed on 2 September 2022).
23. Calvet, J.; Titli, A. Hierarchical Optimisation and Control of Large Scale Systems with Dynamical Interconnection System. *IFAC Proc. Vol.* **1980**, *13*, 117–126. [[CrossRef](#)]
24. Bojović, P.D.; Malbašić, T.; Vujošević, D.; Martić, G.; Bojović, Ž. Dynamic QoS Management for a Flexible 5G/6G Network Core: A Step toward a Higher Programmability. *Sensors* **2022**, *22*, 2849. [[CrossRef](#)] [[PubMed](#)]
25. Yu, Y.; Jiang, X.; Jin, G.; Gao, Z.; Li, P. A Buffer Management Algorithm Based on Dynamic Marking Threshold to Restrain MicroBurst in Data Center Network. *Information* **2021**, *12*, 369. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.