

Hindawi Publishing Corporation
Modelling and Simulation in Engineering
Volume 2012, Article ID 301012, 7 pages
doi:10.1155/2012/301012

Research Article

Average Bandwidth Allocation Model of WFQ

Tomáš Balogh and Martin Medvecký

Institute of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovakia

Correspondence should be addressed to Tomáš Balogh, tomas.balogh@stuba.sk

Received 12 April 2012; Revised 21 October 2012; Accepted 5 November 2012

Academic Editor: Agostino Bruzzone

Copyright © 2012 T. Balogh and M. Medvecký. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new iterative method for the calculation of average bandwidth assignment to traffic flows using a WFQ scheduler in IP based NGN networks. The bandwidth assignment calculation is based on the link speed, assigned weights, arrival rate, and average packet length or input rate of the traffic flows. We prove the model outcome with examples and simulation results using NS2 simulator.

1. Introduction

The current trends in telecommunication infrastructure with packet oriented networks bring up the question of supporting Quality of Service (QoS). Methods, that are able to assign priorities to flows or packets and then service them differently according to their needs in network nodes, were proposed for the demands of QoS support. Queue Scheduling Discipline (QSD) algorithms are responsible for choosing packets to output from queues. They are designed to divide the output capacity fairly and optimally. Algorithms that are able to make this decision according to priorities are the basic component of modern QoS supporting networks [1].

For an optimal configuration of these algorithms we need to calculate or simulate the result of our setting to expect the impact on QoS. The network nodes can be modeled using Markovian models [2].

Most of the existing WFQ bandwidth allocation models do not consider variable utilization of queues or bandwidth redistribution of unassigned link capacity. For this reason we proposed our iterative mathematical model for bandwidth allocation of WFQ. The model can be used for the analysis of the impact of weight settings, analyzing the stability of the system and modeling of delay and queue length of traffic classes.

The next sections of the paper are structured as follows. At first the WFQ algorithm is presented followed by a short

presentation of common used bandwidth constraint models. The third section of the paper describes the proposed model for average bandwidth allocation of WFQ followed by examples of WFQ bandwidth allocation and simulation results proving the proposed model.

2. Bandwidth Allocation

There are many scheduling algorithms and several bandwidth allocation models proposed for bandwidth allocation estimation. We focused on WFQ and bandwidth allocation models proposed for MPLS traffic engineering.

2.1. Weighted Fair Queuing. WFQ was introduced in 1989 by Demers et al. and Zhang [3, 4]. The algorithm provides fair output bandwidth sharing according to assigned weights. The decision which packet should be read from the packet queue and sent next is done by calculating a virtual finish-time. The scheduler assigns the finish time to each packet as it arrives in the queue. The time corresponds with the time, in which the packet would be completely sent bit by bit from each queue as in the Generalized Processor Sharing (GPS) algorithm. The number of bits calculated in one turn corresponds with the assigned weights. The packet with the smallest finish time is chosen for output. WFQ guarantees that each traffic class gets a portion of the output bandwidth and shares it proportional to the assigned weights.

2.2. *Bandwidth Constraint Models.* One of the goals of Diff-Serv or MPLS traffic engineering is to guarantee bandwidth reservations for different service classes. For these goals two functions are defined [5]:

- (i) class-type (CT) is a group of traffic flows, based on QoS settings, sharing the same bandwidth reservation;
- (ii) bandwidth constraint (BC) is a part of the output bandwidth that a CT can use.

For the mapping between BCs and CTs the maximum allocation model (MAM), max allocation with reservation (MAR), and Russian dolls model (RDM) are defined.

Maximum Allocation Model. The MAM model [6] maps one BC to one CT. The whole bandwidth is strictly divided and no sharing between CTs is allowed.

Max Allocation with Reservation. MAR [7] is similar to MAM in that a maximum bandwidth is allocated to each CT. However, through the use of bandwidth reservation and protection mechanisms, CTs are allowed to exceed their bandwidth allocations under conditions of no congestion but revert to their allocated bandwidths when overload and congestion occurs [6].

Russian Dolls Model. The RDM model is more effective in bandwidth sharing. It assigns BCs to groups of CTs. For example CT7 with the highest QoS requirements gets its own BC7. The CT6 with lower QoS requirements shares its BC6 with CT7, and so forth. In extreme cases the lower priorities get less bandwidth as they need or even starve [8].

3. WFQ Bandwidth Allocation Model

In general, WFQ and some other scheduling algorithms like WRR, WF²Q+, and so forth allocate bandwidth differently as the models described in Section 2.2. The available bandwidth is divided between service classes or waiting queues according to assigned weights. The sharing of unused bandwidth is allowed and is divided between the other queues again according to assigned weights.

The proposed model is a part of the research of modelling of traffic parameters of NGN networks, is a modification a presented model for bandwidth allocation of the WRR algorithm, and will be further used for delay and queue length modeling of these algorithms.

3.1. *Definitions and Notations.* We assume a network node with P priority classes or waiting queues. Each queue i has a weight w_i assigned. Packets enter the queue with an arrival rate λ_i and mean packet size L_i . The product of these two variables represents the input bandwidth of the priority:

$$I_i = \lambda_i L_i. \quad (1)$$

The total available output bandwidth T will be divided between the priority classes and each of them will get B_i .

For the bandwidth calculation an iterative method will be used. The k th iteration of B_i will be noted as $B_{i,k}$.

3.2. *Model Proposal.* To describe the bandwidth allocation of WFQ, we have to analyze all possible situations that can occur. We will use an iterative method for the analysis.

Let us take a look at the possible situations that can appear in the first step of bandwidth allocation. The WFQ algorithm works at the principle that a number of bits represented by the weight value are sent at once to a virtual output. The bits are then reorganized to the original packets and the packet which is completely transmitted in this way is dequeued as the first. This assures an exact bandwidth allocation between queues according to assigned weights. The distribution of the available bandwidth can be written as follows:

$$T \frac{w_i}{\sum_{j=1}^P w_j}. \quad (2)$$

After the bandwidth is divided between the queues according to (2), there are 3 possible situations.

- (i) The first possibility is that each queue gets and uses the bandwidth calculated in (2). No additional sharing of unused bandwidth will happen. This will happen if

$$T \frac{w_i}{\sum_{j=1}^P w_j} \leq \lambda_i L_i = I_i, \quad i = 1, 2, \dots, P. \quad (3)$$

- (ii) The second option is that each queue is satisfied with the assigned bandwidth. In this case:

$$T \frac{w_i}{\sum_{j=1}^P w_j} \geq \lambda_i L_i = I_i, \quad i = 1, 2, \dots, P. \quad (4)$$

In these two cases, the bandwidth assignment is finished in the first iteration step. No unused bandwidth needs to be divided between other queues. A queue gets the bandwidth which it needs (1) or the proportion of bandwidth based on the WFQ rules (2):

$$B_{i,1} = \min \left(I_i, T \frac{w_i}{\sum_{j=1}^P w_j} \right). \quad (5)$$

This (5) represents also our first iteration step.

If the conditions (3) or (4) are not met, we have to calculate the bandwidth assignment in the next iteration steps. This means some queues need more bandwidth than has been assigned using (2), some others use only the bandwidth calculated in (1), and the rest of the bandwidth is unused and can be shared. We will reassign the unused bandwidth only between the queues whose requirements are not satisfied. The queues that do not need more bandwidth can be identified as follows:

$$I_i - B_{i,k-1} = 0. \quad (6)$$

If the queues bandwidth requirements are met, the result of (6) will be zero. On the other hand a positive number

indicates that the queue needs more bandwidth. This will help us to identify the queues with enough bandwidth or with bandwidth shortage.

The reallocation of the unused capacity will be done only between the queues whose bandwidth requirements are not satisfied until all capacity is divided or all queue requirements met and can take $P - 1$ steps in the worst case. The next iterative step can be written as follows:

$$B_{i,k} = \left(I_i, B_{i,k-1} + \left(T - \sum_{j=1}^P B_{j,k-1} \right) \times \frac{w_i}{\sum_{j=1}^P w_j \min(I_j - B_{j,k-1}, 1)} \right). \quad (7)$$

Equation (7) will be used for calculation of all other iterations from $k = 2$ to $k = P$. The calculation has to stop after all bandwidth requirements of the queues are met otherwise it leads to division by zero. The conditions for the termination of the calculation are as follows.

- (i) The whole output bandwidth is already distributed between the queues:

$$T = \sum_{i=1}^P B_{i,k}, \quad (8)$$

- (ii) or all the requirements of the queues are satisfied:

$$B_{i,k} = I_i = \lambda_i L_i, \quad i = 1, 2, \dots, P. \quad (9)$$

These conditions are also met if in the next iteration no redistribution of bandwidth occurs:

$$B_{i,k} = B_{i,k-1}, \quad i = 1, 2, \dots, P. \quad (10)$$

4. Analysis of Different Behavior Variants

Let us demonstrate the performance of our model in the comparison with WFQ on some examples. In these examples we will assume 4 priority classes. We will show 4 different behaviors. The first example presents the situation, where all traffic classes get the required bandwidth. The second one shows the case in which the bottleneck link has less capacity than is needed and the distribution is done according to packet size and weights. The third example shows us the worst case in which redistribution of bandwidth occurs and the calculation of bandwidth takes P iterations. The last example demonstrates also bandwidth reallocation but the reallocation process will stop after less than P iterations.

Example 1. In this example we will assume a 100 Mbps output link. The first class represents a VoIP flow with high traffic. The mean packet size is set to 100 B what equals to $L_1 = 800$ bits. The packets enter the system with a mean inter packet interval 10 ms, which represents an arrival rate of $\lambda_1 = 100$. The second class represents a video conference

with $L_2 = 8000$ bits and $\lambda_2 = 10$. The third class represents video streaming with the same parameters. The last fourth class transports data with lowest priority settings. The traffic parameters are $L_4 = 12000$ bits and $\lambda_4 = 1$.

The input bandwidths calculated using (1) are $I_1 = 80$ kbps, $I_2 = 80$ kbps, $I_3 = 80$ kbps, and $I_4 = 12$ kbps.

The weights are set in the following way: $w_1 = 4$, $w_2 = 2$, $w_3 = 2$, and $w_4 = 1$. The bandwidth allocated to the queues according to (2) is 40 Mbps, 30 Mbps, 20 Mbps, and 10 Mbps, which is more than all the queues need. In this case the iterations are stopped after the first step (5) and the bandwidth used by the queues is the lower value of this equation I_i . We stopped the iterations according to the condition defined in (9).

Example 2. This example uses the same traffic settings as in Example 1. The only difference is that the output link capacity is set to 50 kbps.

The bandwidth allocation calculated using (2) is 20 kbps, 15 kbps, 10 kbps, and 5 kbps. This represents the whole 50 kbps output capacity. None of the traffic classes has enough capacity to redistribute and the bandwidth allocation is done again in the first iteration.

Example 3. In this example we will show the worst case in which the bandwidth allocation stops after the maximal P steps. We will use the same packet size $L_i = 375$ B in all queues. The weights are again set as follows: $w_1 = 4$, $w_2 = 2$, $w_3 = 2$, and $w_4 = 1$. There are different arrival rates that are set to modify the required bandwidth and present the reallocation. The arrival rates are set to $\lambda_1 = 1000$, $\lambda_2 = 1041.667$, $\lambda_3 = 1000$, and $\lambda_4 = 416.667$. The output link capacity is set to 10 Mbps.

This settings result into the following bandwidth requirements calculated using (1): 3 Mbps, 3.125 Mbps, 3 Mbps, and 1.25 Mbps, where the sum of these bandwidths is higher than the output capacity. All the bandwidth calculations are also visible in Table 1.

In the first iteration the bandwidth allocated using (2) is 4 Mbps, 3 Mbps, 2 Mbps, and 1 Mbps. The first traffic class can use only 3 Mbps of the assigned capacity and the remaining 1 Mbps is divided between the remaining 3 classes. This result corresponds with the proposed model (5).

In the second iteration the result of (6) is equal to zero for the first flow which means that the remaining capacity will be divided between classes 2, 3, and 4. The remaining 1 Mbps is divided again according to the weights in the following way: 0.5 Mbps, 0.333 Mbps, and 0.167 Mbps and added to the already assigned bandwidth.

In the 3rd iteration the remaining capacity of 0.375 Mbps is divided between classes 3 and 4 by the ratio of 2 : 1 due to the assigned weights. This capacity is added to the previously assigned and results into 3 Mbps, 3.125 Mbps, 2.583 Mbps, and 1.292 Mbps.

In the 4th and last reallocation of bandwidth the unused capacity 0.042 Mbps of class 4 is reassigned to the last unsatisfied class 3 and fully used. The resulting allocation of bandwidth is as follows: 3 Mbps, 3.125 Mbps, 2.625 Mbps, and 1.25 Mbps.

TABLE 1: Allocation of bandwidth in Example 3.

Traffic class	Required (Mbps)	Bandwidth						
		1st iteration		2nd iteration		3rd iteration		4th iteration
		Allocated (Mbps)	Unallocated (Mbps)	Allocated (Mbps)	Unallocated (Mbps)	Allocated (Mbps)	Unallocated (Mbps)	Allocated (Mbps)
Traffic class 1	3	4	1	3		3		3
Traffic class 2	3.125	3		3.5	0.375	3.125		3.125
Traffic class 3	3	2		2.333		2.583		2.625
Traffic class 4	1.25	1		1.167		1.292	0.045	1.25

All these results correspond with the proposed models (5) and (7).

Example 4. This example describes the bandwidth allocation, where the calculation has to be stopped after the conditions in (8) or (9) are met.

The weight and packet size settings are the same as in the previous Example 3. The output bandwidth is set again to 10 Mbps. The arrival rates are set to 1000, 1000, 750, and 500 pps (packets per second). These settings lead to bandwidth requirements of 3, 3, 2.25, and 1.5 Mbps as a result of (1).

In the first iteration using (2) we allocate 4, 3, 2, and 1 Mbps to the queues. In this case the first queue has 1 Mbps remaining for reallocation and the second queue is already satisfied with the allocated bandwidth.

The second iteration reassigns the 1 Mbps divided using the ratio 2:1 to queues 3 and 4. The bandwidth assigned to them is 2.667 Mbps and 1.333 Mbps, but queue 3 needs only 2.25 Mbps output capacity and the remaining part of the capacity can be reassigned to the last unsatisfied queue 4.

In the third iteration we assign 3 Mbps to the first queue, 2 Mbps to the second queue, 2.25 Mbps to the third queue, and 1.75 Mbps to the last queue. The fourth queue needs only 1.5 Mbps and this means that bandwidth requirements of all queues are met. The iterations have to stop at this moment according to (9) otherwise the model would lead to dividing by zero.

We can change the arrival rate of the fourth queue to 750 pps and raise the bandwidth requirements 2.25 Mbps. In this case in the 3rd iteration the bandwidth allocations are 3, 3, 2.25, and 1.75 Mbps. This means that the whole output capacity is divided to the queues (8) and we can stop the iteration. Otherwise each next step would lead to same results.

5. Simulations

To prove the results of our mathematical model we used simulations in the NS2 simulation software [9] (version 2.29) with DiffServ4NS patch [10].

For the simulations a simple network model with 4 transmitting nodes (1–4) and four receiving nodes (6–9) was used. The transmitting and receiving nodes are interconnected with one link between nodes 0 and 5. The node 0 uses WFQ to schedule packets on this bottleneck link

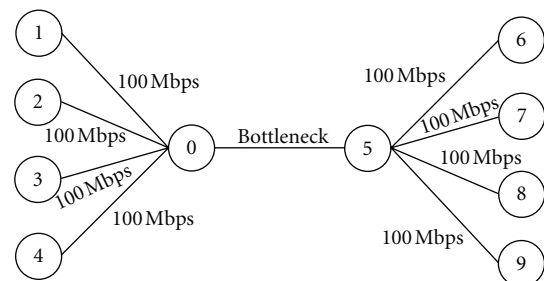


FIGURE 1: Simulation model.

where the mentioned bandwidths are set. All other links have a capacity of 100 Mbps. The model is shown in Figure 1. The queues at node 0 have enough capacity so no packet loss will occur.

We used two types of traffic sources. The first one generates packets only with one packet size and constant packet interval. These settings are easier to simulate and represent a D/D/1/∞ Markovian model.

The second traffic source type represents an M/M/1/∞ model. There is a lack of possibility to generate traffics with different packet sizes in NS2 simulator. For this reason the M/M/1 source is modeled using an ON/OFF source where each node generates one packet with a random size (exponential distribution) and the interval for the next packet transmission is a random time (again a random number with exponential distribution).

An example of input data generated at one node with the mean packet size 375 B and arrival rate 1000 pps is shown in Figures 2 and 3. The red line represents the number of packets generated corresponding with the exponential probability calculated for these settings and the blue bars represent the histogram of packets generated in the simulation that lasted 100 s.

We made many simulations under different parameter settings. The presented results correspond with described examples or present different extreme settings. The results of simulations of M/M/1 and D/D/1 models and the results of our proposed model are shown in Table 2.

We measured the bandwidth after achieving “steady state.” The measurement started after 20 s of simulation when the bandwidth was stable and queues filled up with waiting packets [11].

TABLE 2: Simulation results compared with mathematical model results.

Simulation variant (#)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Mean packet size (B)	100, 1000, 1000, 1500	100, 1000, 1000, 1500	375, 375, 375, 375	375, 375, 375, 375	375, 375, 375, 375	1000, 100, 10, 1	1000, 1000, 1000, 1000	1000, 100, 1000, 100	1000, 1000, 1000, 1000
Mean arrival rate (pps)	100, 10, 10, 1	100, 10, 10, 1	1000, 1041.67, 1000, 416.67	1000, 1000, 750, 500	1000, 1000, 750, 750	100, 100, 100, 100	1, 10, 100, 1000	1000, 100, 1000, 100	100, 100, 100, 100
Input bandwidth (Mbps)	0.08, 0.08, 0.08, 0.012	0.08, 0.08, 0.08, 0.012	3, 3, 1.25, 3, 1.25	3, 3, 2.25, 1.5	3, 3, 2.25, 2.25	0.8, 0.08, 0.008, 0.0008	0.008, 0.08, 0.8, 8	8, 0.08, 8, 0.08	0.8, 0.8, 0.8, 0.8
Weight settings	4, 3, 2, 1	4, 3, 2, 1	4, 3, 2, 1	4, 3, 2, 1	4, 3, 2, 1	1, 1, 1, 1	4, 3, 2, 1	4, 3, 2, 1	40, 3, 2, 1
Link capacity (Mbps)	100	0.05	10	10	10	0.5	4	8	1.6
D/D/1 simulation results (Mbps)	0.079, 0.079, 0.079, 0.012	0.019, 0.015, 0.01, 0.005	3.00, 3.15, 2.59, 1.248	2.99, 2.99, 2.249, 1.50	2.99, 3.00, 2.25, 1.749	0.41, 0.08, 0.0079, 0.0008	0.0078, 0.08, 0.8, 3.111	5.226, 0.079, 2.6133, 0.079	0.79, 0.4, 0.267, 0.133
M/M/1 simulation results (Mbps)	0.0804 ± 0.47%, 0.0801 ± 1.34%, 0.0802 ± 1.26%, 0.0122 ± 1.69%	0.0200 ± 0.06%, 0.0150 ± 0.10%, 0.010 ± 0.12%, 0.0050 ± 0.36%	3.0041 ± 0.13%, 3.1296 ± 0.14%, 2.6134 ± 0.30%, 1.2529 ± 0.27%	3.0061 ± 0.43%, 3.0022 ± 0.23%, 2.2501 ± 0.32%, 1.5007 ± 0.33%	3.0054 ± 0.34%, 3.0030 ± 0.13%, 2.2564 ± 0.21%, 1.7352 ± 0.66%	0.4095 ± 0.21%, 0.0808 ± 1.07%, 0.0085 ± 1.01%, 0.0013 ± 0.77%	0.0080 ± 2.73%, 0.0799 ± 0.97%, 0.7991 ± 0.67%, 3.1129 ± 0.17%	5.2261 ± 0.01%, 0.0804 ± 0.42%, 2.6131 ± 0.07%, 0.0805 ± 0.57%	0.8015 ± 0.56%, 0.3993 ± 0.57%, 0.2662 ± 0.57%, 0.1331 ± 0.56%
Model results (Mbps)	0.08, 0.08, 0.08, 0.012	0.02, 0.015, 0.01, 0.005	3, 3, 1.25, 2.625, 1.25	3, 3, 2.25, 1.5	3, 3, 2.25, 1.75	0.411, 0.08, 0.008, 0.0008	0.008, 0.08, 0.8, 3.112	5.2266, 0.08, 2.6133, 0.08	0.8, 0.4, 0.266, 0.133

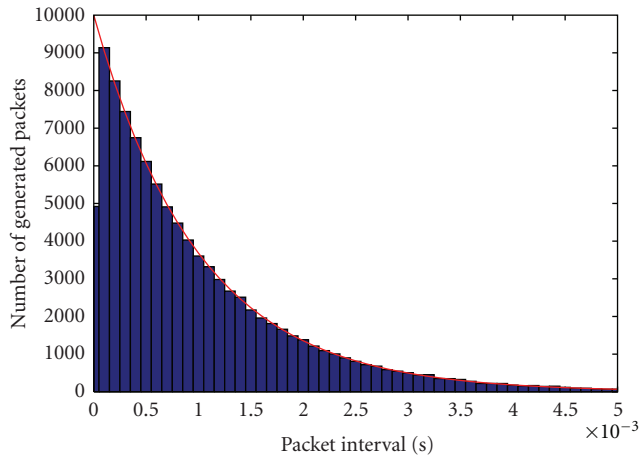


FIGURE 2: Exponential probability distribution of arrival rate with mean value 1000 pps.

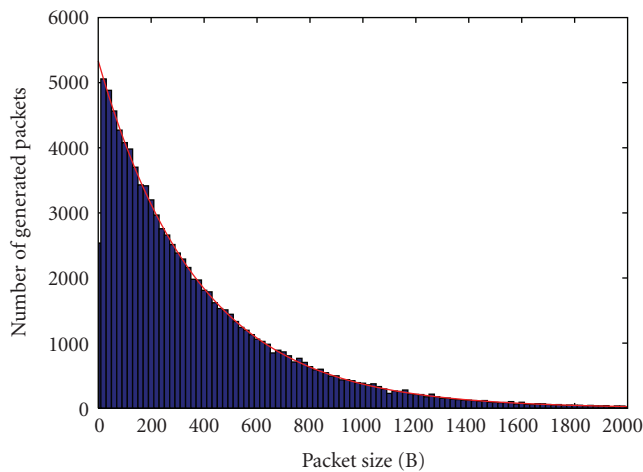


FIGURE 3: Exponential probability distribution of packet sizes with mean value 375 B.

The results of the mathematical model mostly correspond with the simulation results. The results of the D/D/1 simulation model are more exact due to the exact setting of packet size. The small inaccuracy can be caused by measurement errors, where the bandwidth calculation is stopped closely to an arrival of a packet when small arrival rates are set. Due to the deterministic parameter settings there is no difference between more runs of simulations and no result variance occurs.

The presented results for the M/M/1 simulations are an average value calculated from 10 simulation runs and the standard deviation of the runs is also provided. The simulation runs for most parameter settings lasted 200 s. In cases an extreme low arrival rate was set we extended the simulation duration up to 1000 s. We provided also simulations with WF²Q+ [12] scheduler instead of WFQ. The simulation results correspondent with the presented results and proved that this model is applicable also for other WFQ based schedulers that use packet size for the dequeue order decision.

6. Conclusion

We presented a new iterative bandwidth allocation model for WFQ in IP based NGN networks. The proposed model uses the weight settings of the WFQ scheduler and average input bandwidth of different flows for the bandwidth calculation. The variable utilization of different queues and packet redistribution is considered in the calculations. The proposed model allows to easily predict the impacts of the scheduler, traffic shapers, and input traffics on QoS of the transported data.

The functionality of the model was presented on five different examples and confirmed by simulations in the NS2 simulator for both D/D/1 and M/M/1 input traffics.

The proposed iterative bandwidth allocation model was tested with WF²Q+ scheduler with the same simulation results. Therefore we can say that proposed model is also applicable on other WFQ based schedulers.

The results of this bandwidth allocation model will be used in further research of delay and packet loss modeling using Markovian queue models.

Acknowledgments

This work is a part of research activities conducted at Slovak University of Technology Bratislava, Faculty of Electrical Engineering and Information Technology, Institute of Telecommunications, within the scope of the project “Support of Center of Excellence for SMART Technologies, Systems and Services II., ITMS 26240120029, cofunded by the ERDF.”

References

- [1] T. Mišuth and I. Baroňá, “On queueing systems application in IP networks,” in *EE Časopis Pre Elektrotechniku A Energetiku*, vol. 16, pp. 91–94, 2010.
- [2] T. Mišuth, E. Chromý, and M. Kavacký, “Prediction of traffic in the contact centers,” in *Proceedings of the 6th International Conference on Electrical and Electronics Engineering (ELECO’09)*, pp. 111–114, Bursa, Turkey, November 2009.
- [3] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” in *Proceedings of the ACM Computer Communication Review (SIGCOMM ’89)*, pp. 3–12, 1989.
- [4] L. Zhang, “Virtual clock: a new traffic control algorithm for packet switching networks,” in *Proceedings of the ACM Transactions on Computer Systems*, vol. 9, no. 2, pp. 101–124, 1990.
- [5] K. Molnar and M. Vlcek, “Evaluation of bandwidth constraint models for MPLS networks,” *Electronics*, vol. 1, pp. 172–175, 2009.
- [6] RFC, 4125, *Maximum Allocation Bandwidth Constraints Model for DiffServ-Aware MPLS Traffic Engineering*, The Internet Society, 2005.
- [7] RFC, 4126, *Max Allocation with Reservation Bandwidth Constraints Model for Diffserv-Aware MPLS Traffic Engineering & Performance Comparisons*, The Internet Society, 2005.
- [8] RFC, 4127, *Russian Dolls Bandwidth Constraints Model for DiffServ-Aware MPLS Traffic Engineering*, The Internet Society, 2005.

- [9] The Network Simulator-ns-2, <http://www.isi.edu/nsnam/ns/>.
- [10] Androzzzi, "Sergio. DiffServ4NS," <http://sergioandrozzzi.com/research/network/diffserv4ns/>.
- [11] K. Pawlikowski, H. D. J. Jeong, and J.S. R. Lee, "On credibility of simulation studies of telecommunication networks," *IEEE Communications Magazine*, vol. 40, no. 1, pp. 132–139, 2002.
- [12] J. C. R. Bennet and H. Zhang, "Hierarchical packet fair queuing algorithms," in *Proceedings of the IEEE/ACM Transactions on Networking*, pp. 675–689, 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

