

Average Delay and Queue Length Model for WRRPQ

TOMÁŠ BALOGH, MARTIN MEDVECKÝ

Institute of Telecommunications

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in
Bratislava

Ilkovičova 3, 812 19 Bratislava

SLOVAKIA

tomas.balogh@stuba.sk, martin.medvecky@stuba.sk

Abstract: - We present a new iterative mathematical model for calculating the average bandwidth assigned to traffic flows using a Weighted Round Robin with Priority queuing (WRRPQ) scheduler with both strict and rate controlled priority in IP based NGN networks. The bandwidth assignment estimation is based on the average packet length, link speed and the arrival rate. We prove the model outcome with examples and simulation results using NS2 simulator.

Key-Words: - Bandwidth Constraint Model, IP, NGN, WRRPQ, PQ, WRR

1 Introduction

With the technology transfer to new telecommunication infrastructure concepts, which uses packet oriented networks, brings up the question of supporting Quality of Service (QoS). Methods, which 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 optimal. Algorithms that are able to make this decision according to priorities are the basic component of modern QoS supporting networks[1].

The increase of data transfer and link capacity results in a search of new algorithms, which are easier to implement, but were not designed for IP networks as Weighted Round Robin (WRR). For the correct implementation and configuration of these algorithms we need to model and analyze them and modify their settings to achieve the behavior as more complex algorithms like Weighted Fair Queuing (WFQ).

Most of the existing 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 WRRPQ. The model can be used for the analysis of the impact of weight settings and rate limit,

analyzing the stability of the system and modeling the assigned bandwidth.

For an optimal configuration of these algorithms we need to calculate or simulate the result of our setting to expect the impact on QoS [2].

The next sections of the paper are structured as following. First the QSD algorithms WRR and WRRPQ are presented followed by a short presentation of common used Bandwidth constraint models. The fourth section of the paper describes the proposed model for average bandwidth allocation of the WRRPQ algorithm followed by examples and simulation results proving the proposed model.

2 Queue Scheduling Algorithms

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

2.1 Weighted Round Robin

Round Robin is a well-known QSD and also used for process sharing in operation systems. It assigns to each queue equal service based on number of packets without priorities and calculation of packet size. Therefore are RR based algorithms more suitable for ATM networks with constant packet-size.

WRR [3] extends the RR algorithm by a possibility to assign different weights to queues. It allows differentiating in service class handling. Packets are first classified into service classes and assigned to specified queues. Each queue is visited by the scheduler and packets are sent from the queue.

There are two ways how to implement different weights for queues [4]:

- Sending multiple packets during one visitation of the scheduler. The number of packets corresponds with assigned weight.
- Multiple visits of the scheduler according to the weight of service class.

WRR queuing can be implemented in hardware, so it has lower computing requirements and can be used in high-speed nodes in network. The visitation and choosing packets for output from each queue ensures all services to get some portion of the output capacity. It prevents starvation.

The output capacity is allocated according to number of packets. Their size is not calculated. This means only rough control over the output bandwidth allocated to each queue.

When one queue is empty, WRR divides the bandwidth allocated to that queue is divided to the remaining queues according to their weights.

2.2 Weighted Round Robin with Priority queuing

Weighted Round Robin with Priority queuing (WRRPQ) [5] is a combination of Priority Queuing (PQ) and WRR.

Priority queuing [6] is a simple queue scheduling algorithm to provide simple method of differentiated service classes. Packets are dequeued from the head of a queue only if all queues with higher priority are empty. The simplicity allows implementation with low computation requirements.

Disadvantage of this algorithm is that packets of lower priority may experience high delay or even starvation if high priority traffic is not policed or conditioned at the edge of network.

The combination of both algorithms provides low delay, packet loss and jitter for high priority queues by supplying them strict priority. If the high priority class doesn't exceed the available output capacity other classes are serviced as classical WRR.

There is also the possibility to limit the priority queue to a fixed rate. Then the queue serviced using PQ will use the bandwidth required by the traffics assigned to this queue up to this limit. Other packets

will remain waiting in the queue and other queues will be serviced using WRR. When setting an accurate value of the limit we can avoid starvation of lower priority queues.

3 Bandwidth Constraint Models

One of the goals of DiffServ or MPLS traffic engineering is to guarantee bandwidth reservations for different service classes. For these goals two functions are defined [7]:

- Class-type (CT) is a group of traffic flows, based on QoS settings, sharing the same bandwidth reservation.
- 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.

3.1 Maximum Allocation Model

The MAM model [8] maps one BC to one CT. The whole bandwidth is strictly divided and no sharing between CTs is allowed.

3.2 Max Allocation with Reservation

MAR [9] 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 [8].

3.3 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 etc. In extreme cases the lower priorities get less bandwidth as they need or even starve [10].

4 WRRPQ Bandwidth Allocation Model

WRRPQ and some other scheduling algorithms like WFQ, WF²Q+, etc. allocate bandwidth different as

the models described above. The available bandwidth is divided between service classes or waiting queues according to the assigned weights. The sharing of unused bandwidth is allowed and is divided between the other queues again according to assigned weights.

The WRRPQ assigns bandwidth according the principle of priority queuing for the first queue and according to weights for the remaining queues. We have to take the packet length into account when calculating the bandwidth allocation. The reason for this is that the algorithm was originally designed for ATM networks with constant packets.

4.1 Definitions and notations

We assume a network node with P priority classes or waiting queues. The first queue has no weight assigned. It is possible to assign a rate limit R_1 to this queue. Other queues i have a weight w_i assigned. Packets, with the mean packet size L_i , enter the queue with the arrival rate λ_i . 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 $B_{i,k}$.

4.2 Model proposal

Each of our inputs will produce a mean input bandwidth which is equal to:

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

We have to analyze all possible situations which can occur to correctly describe the algorithm mathematically. We will use an iterative method for description of the algorithm mentioned above.

Let us take a look at the possible situations that can appear in the first step of bandwidth allocation.

As first the first queue served with strict priority queuing will get the required bandwidth assigned up to the capacity of the output link. No bandwidth reallocation will occur in the next iterations for this queue. Therefore we can write:

$$B_{1,1} = B_{1,2} = \dots = B_{1,P} = B_1 = \min(\lambda_1 L_1, T) \quad (2)$$

If we use the variant of PQ with rate-controlled priority the first queue gets the bandwidth assigned which equals to the input bandwidth requirements up to the rate limit:

$$B_{1,1} = B_{1,2} = \dots = B_{1,P} = B_1 = \min(\lambda_1 L_1, R_1, T) \quad (3)$$

The next queues are served as classical WRR and the calculation is similar to the presented in [10]. The bandwidth assignment continues with the other queues only if there is some remaining capacity after the assignment to the first queue:

$$T > B_1 \quad (4)$$

WRR algorithm works at the principle that a number of packets represented by the weight value is sent at once to output and assigns to each queue a part of the remaining output bandwidth according to assigned weight and average packet length in proportion to other queues:

$$(T - B_1) \frac{w_i L_i}{\sum_{j=1}^P w_j L_j} \quad (5)$$

The first possibility is that each queue gets and uses this bandwidth. No additional sharing of unused bandwidth will occur. This will happen if:

$$(T - B_1) \frac{w_i L_i}{\sum_{j=1}^P w_j L_j} \leq \lambda_i L_i \quad i = 2, \dots, P \quad (6)$$

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

$$(T - B_1) \frac{w_i L_i}{\sum_{j=1}^P w_j L_j} \geq \lambda_i L_i \quad i = 2, \dots, P \quad (7)$$

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 WRR rules (2):

$$B_{i,1} = \min \left(\lambda_i L_i, (T - B_1) \frac{w_i L_i}{\sum_{j=1}^P w_j L_j} \right) \quad (8)$$

This equation (8) represents also our first iteration step.

If the conditions (6) or (7) are not met, we have to calculate the next iteration steps. This means some queues need more bandwidth than assigned using the equation (5) and some other use only (1) of the bandwidth and the rest is unused and can be shared. We will reassign the unused bandwidth only between the queues which requirements are not satisfied. The queues that do not need more bandwidth can be identified as following:

$$\lambda_i L_i - B_{i,k-1} \quad (9)$$

If the queues bandwidth requirements are met, the result of (9) will be zero. On the other hand a positive number indicates that the queue needs more bandwidth. This will help us to identify the requirements.

The reallocation of the unused capacity will be done only between the queues which 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 following:

$$B_{i,k} = \min \left(\lambda_i L_i, B_{i,k-1} + \left(T - \sum_{j=1}^P B_{j,k-1} \right) \frac{w_i L_i}{\sum_{j=2}^P w_j L_j} \cdot \begin{cases} 0 & \text{if } \lambda_j L_j - B_{j,k-1} = 0 \\ 1 & \text{else} \end{cases} \right). \quad (10)$$

The equation (10) will be used for calculation of all other iterations from $k = 2$ to $k = P - 1$. The calculation has to stop after all bandwidth requirements of the queues are met and otherwise it leads to division by zero. The conditions for the termination of the calculations are:

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

$$T = \sum_{i=1}^P B_{i,k}, \quad (11)$$

- or all the requirements of the queues are satisfied:

$$B_{i,k} = \lambda_i L_i \quad i = 2, \dots, P. \quad (12)$$

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

$$B_{i,k} = B_{i,k-1} \quad i = 1, 2, \dots, P. \quad (13)$$

5 Examples

Let us demonstrate our model and the behavior on some examples. We will assume 4 priority classes in these examples. We will show 6 different behaviors. The first three examples present the behavior, where all classes gets the bandwidth they need or all output bandwidth is distributed during the first iteration step. The fourth 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.

5.1 Example I.

In the first example we will use WRRPQ with strict priority. We set the output bandwidth to 10 Mbps. The weights assigned to the queues are $w_2 = 3$, $w_3 = 2$ and $w_4 = 1$ for the last queue. The input traffics have the packet sizes of $L_1 = 1000 B$, $L_2 =$

$500 B$, $L_3 = 375 B$ and $L_4 = 1250 B$. The arrival rates are set to $\lambda_1 = 1250$, $\lambda_2 = 1000$, $\lambda_3 = 1000$ and $\lambda_4 = 300$. Using equation (1) we can calculate the input bandwidths as $I_1 = 10 Mbps$, $I_2 = 4 Mbps$, $I_3 = 3 Mbps$ and $I_4 = 3 Mbps$.

As we use strict priority we will use the equation (2) to calculate the bandwidth assigned to the first queue. The result will be 10 Mbps what is the whole output link capacity and we can stop the calculation as no more bandwidth is available to be assigned to other queues.

The bandwidth allocation would end with the same result either if we would change the input rates, packet sizes of flows 2, 3 and 4 or if would change the bandwidth needs of the flow 1 to higher. The allocation would remain the same as only the capacity of the link can be used for the first queue.

5.2 Example II.

In this example we will use the version of WRRPQ with rate control. The limit for the first queue is set to 4 Mbps. The other settings will remain the same as in *Example I*.

Due to the rate controlled priority we will use equation (3) for the calculation of bandwidth assigned to the first queue. The queue has the input rate of 10 Mbps but due to the limit it will get only 4 Mbps what is also the result of the equation.

After this assignment 6 Mbps are unused and available for the other queue. We will use the equation (8) for the bandwidth assignment. The result of the part of the equation marked as (5) is approximately 2.571 Mbps, 1.286 Mbps and 2.143 Mbps. As it is less than the bandwidth required by all of the queues there is no bandwidth left for redistribution and the calculation can end. The results of (3) and (5) are the bandwidths assigned in this scenario.

5.3 Example III.

In this example the bandwidth will be also allocated in the first iteration to all queues. All of them will get the bandwidth they need to satisfy their needs.

We will use the same traffic parameters as in examples *I*. and *II*. except of the first queue. There the settings will be 400 B for the packet size and the input rate $\lambda_1 = 1250$. The input bandwidth according to (1) will be 4 Mbps. In this example we will also change the output link capacity to 20 Mbps. The

algorithm used is WRRPQ with rate control and the limit is set to 5 Mbps.

The first queue gets all the bandwidth needed according to equation (3) what is 4 Mbps. The remaining capacity of 16 Mbps can be distributed between the other queues.

Using the equation (5) we would assign the capacity of approximately 6.857 Mbps, 3.429 Mbps and 5.714 Mbps what is in all cases more than the required 4, 3 and 3 Mbps. We can stop the calculation now and all the queues will get the bandwidth assigned they need what equals to 4, 4, 3 and 3 Mbps

5.4 Example IV.

In this example we will show the worst case in which the bandwidth allocation stops after P steps. For easier imagination we will use the same packet size $L_i = 375 B$ in all queues. The weights are again set as following: $w_2 = 2$, $w_3 = 2$ and $w_4 = 1$. There are different arrival rates 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. We will use WRRPQ with rate controlled priority and the limit will be set to 4 Mbps.

This settings result into the following bandwidth requirements calculated using (1): 3 Mbps, 3.125 Mbps, 3 Mbps and 1.25 Mbps, what is more than the output capacity.

To the first queue is the bandwidth assigned equal to their input bandwidth 3 Mbps what is less than the limit and also the capacity of the link. This corresponds with the result of (3).

The remaining capacity of 7 Mbps will be distributed between the 3 queues using equation (8). Using the equation (5) the queues get the proportion of the bandwidth calculated using the weights and packet size of 3.5 Mbps, 2.333 Mbps and 1.1667 Mbps. The second queue uses only 3.125 Mbps of the assigned bandwidth and the remaining capacity of 0.375 Mbps can be redistributed in the second iteration. The result of the bandwidth allocation of the first iteration is 3 Mbps, 3.125 Mbps, 2.333 Mbps and 1.1667 Mbps.

In the second iteration we can redistribute the remaining capacity of 0.375 Mbps between the 2 queues (3 and 4) their requirements are still not met. We can identify the using equation (9). The bandwidth will be divided using the ratio 2:1 as the mean packet size is the same for both queues. They get additionally to the bandwidth assigned in the previous iteration the bandwidths of 0.25 and 0.125 Mbps. The fourth queue has now 1.2916 Mbps assigned but uses only 1.25. The remaining capacity will be assigned in the next iteration. The result of this iteration is calculated using (10): 3, 3.125, 2.5833 and 1.25 Mbps.

In the third and last iteration we add the capacity 0.0416 Mbps to the last queue that has higher requirements as the bandwidth already assigned. This is the third queue. The result of the bandwidth allocation model and the 3rd iteration is 3, 3.125, 2.625 and 1.25 Mbps. We stop the calculation now as the maximal number of P-1 was achieved and the whole output bandwidth was distributed between the queues.

All the bandwidth calculations are also visible in Table I.

TABLE I
ALLOCATION OF BANDWIDTH IN EXAMPLE III.

| | Required bandwidth | 1 st iteration | Unused capacity after 1 st iteration | 2 nd iteration | Unused capacity after 2 nd iteration | 3 rd iteration |
|------------------------|--------------------|---------------------------|---|---------------------------|---|---------------------------|
| Traffic class 1 [Mbps] | 3 | 3 | | 3 | | 3 |
| Traffic class 2 [Mbps] | 3.125 | 3.5 | 0.375 | 3.125 | | 3.125 |
| Traffic class 3 [Mbps] | 3 | 2.33333 | | 2.58333 | | 2.625 |
| Traffic class 4 [Mbps] | 1.25 | 1.16667 | | 1.291667 | 0.041667 | 1.25 |

5.5 Example V.

This example shows us one of the options when the bandwidth allocation calculation has to stop after less than P-1 steps. In this case due to the fulfillment of the needs of all queues.

We will use the same packet sizes as in *Example IV*. (375 B for all queues). The arrival rates will be set to $\lambda_1 = 1000$, $\lambda_2 = 1000$, $\lambda_3 = 750$ and $\lambda_4 = 500$. The results of these settings are the bandwidth requirements 3 Mbps, 3 Mbps, 2.25 Mbps and 1.5 Mbps. The weight settings are again $w_2 = 2$, $w_3 = 2$ and $w_4 = 1$ and the output link capacity 10 Mbps.

We can set the queue limit for the priority queue to 4 Mbps but this will give as the same result as if we use WRRPQ with strict priority as the limit is higher as the bandwidth requirements of the queue. Therefore the first queue will get the capacity of 3 Mbps as calculated using (2) or (3).

In the first iteration we can assign the bandwidth of 7 Mbps. According to the equation (5) the queues should get 3.5 Mbps, 2.333 Mbps and 1.1667 Mbps. The second queue uses only 3 Mbps of this capacity and third only 2.25 Mbps. This is also the result of (8).

In the second iteration we can redistribute the remaining capacity of 0.5833 Mbps to the last unsatisfied queue No. 3. This queue gets now 1.75 Mbps but uses only 1.5 Mbps. The requirements of all queues are now fulfilled and the final bandwidth assignment is the same as the input bandwidths of the queues.

We must stop the calculations now as the condition (12) is met, otherwise the model would end with an error as no queues need more bandwidth and the equation (9) is zero for all queues what would mean that the equation (10) has to divide by zero

5.6 Example VI.

The last example also presents a situation where we can stop the calculations bin less than P-1 steps, in this case due to the distribution of the whole output capacity. We will use the same setting as in the previous example. The only difference is the arrival rate of the fourth queue $\lambda_4 = 750$.

The first iteration has the same results as in the *Example V*. In the second iteration we assign also the remaining capacity of 0.5833 Mbps to the last unsatisfied queue No. 3. The bandwidth requirements of this queue are 2.25 Mbps what is more than the assigned capacity of 1.75 Mbps.

In the third iteration we have no more bandwidth to assign as the whole output capacity is already used. The condition (11) is met and all other iterations would end with the same result.

6 Simulations

To prove the results of our mathematical model we used simulations in the NS2 simulation software [12] (version 2.29) with DiffServ4NS patch [13].

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 our own implementation of WRRPQ algorithm with both strict priority and also rate limited priority queue to schedule packets on this bottleneck link where the mentioned bandwidths are set. All other links have a capacity of 1000 Mbps. The model is shown in Fig. 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).

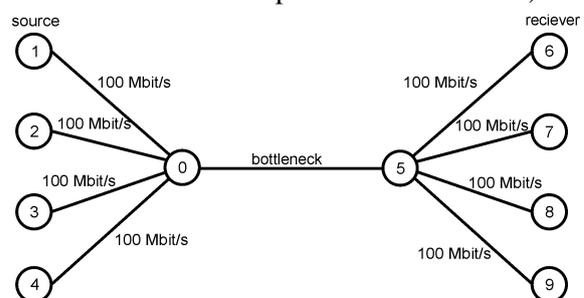


Fig. 1: Simulation model

An example of input data generated at one node with the mean packet size 375 B and arrival rate 1000 pps is shown in Fig. 2 and Fig. 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.

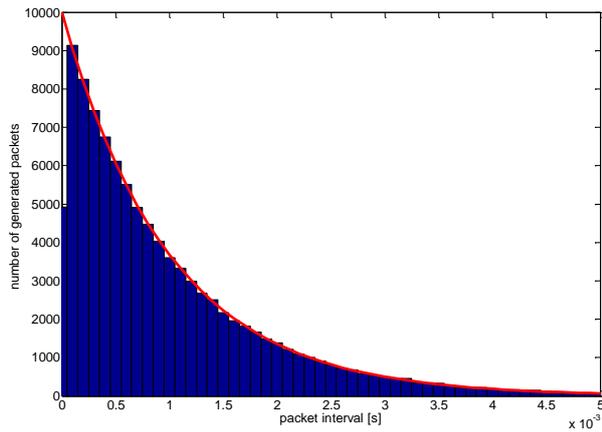


Fig. 2: Exponential probability distribution of arrival rate with mean value 1000 pps

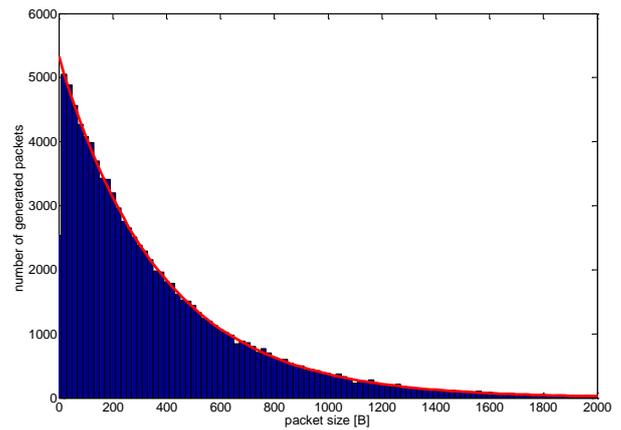


Fig. 3: Exponential probability distribution of packet sizes with mean value 375 B

TABLE II
SIMULATION RESULTS I.

| Simulation variant [#] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---------------------------------|---|---|--|--|--|--|---|--|---|--|-------------|
| Mean packet size [B] | 1000 500 375 1250 | 1000 500 375 1250 | 400 500 375 1250 | 375 375 375 375 | 375 375 375 375 | 375 375 375 375 | 375 375 375 375 | 375 375 375 375 | 1000 100 1000 100 | 1000 100 1000 100 | |
| Mean arrival rate [packets/s] | 1250 1000 1000 300 | 1250 1000 1000 300 | 1250 1000 1000 300 | 1000 1041.67 1000 416.667 | 1000 1000 750 500 | 1000 1000 750 750 | 1000 1000 750 750 | 1000 1000 750 750 | 1000 100 1000 100 | 1000 100 1000 100 | |
| Weight settings | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | |
| Link capacity [Mbps] | 10 | 10 | 20 | 10 | 10 | 10 | 10 | 10 | 8 | 8 | |
| Priority queue limit [Mbps] | - | 4 | 5 | 4 | 4 | 4 | 2.8 | - | 4 | - | |
| Model results [Mbps] | 10 0 0 | 4 2,571 1,286 2,143 | 4 4 3 | 3 3,125 2,625 1,25 | 3 3 2,25 1,5 | 3 3 2,25 1,75 | 2,8 3 2,25 1,95 | 2,8 3 2,25 1,75 | 3 3 2,25 1,75 | 4 0,08 3,84 0,08 | 8 0 0 |
| D/D/1 simulation results [Mbps] | 9,998 0,000827 0,00042 0,0007 | 4,001 2,57 1,285 2,142 | 4,000 4,003 3,000 3,002 | 3,000 3,128 2,621 1,251 | 3,000 3,002 2,250 1,501 | 3,002 3,001 2,251 1,746 | 2,801 3,000 2,251 1,948 | 2,999 3,000 2,251 1,749 | 4,000 0,080 3,840 0,080 | 7,994 0,001 0,005 0,000 | |
| M/M/1 simulation results [Mbps] | 9,999±0.13% 0,003±200% 0,001±199% 0,002±194% | 4,002±0.001% 2,576±0.13% 1,290±0.17% 2,130±0.14% | 4,004±0.19% 4,002±0.27% 3,005±0.21% 3,003±0.63% | 3,007±0.26% 3,133±0.29% 2,609±0.34% 1,250±0.45% | 3,002±0.2% 3,005±0.29% 2,251±0.37% 1,501±0.4% | 3,005±0.26% 3,003±0.14% 2,255±0.26% 1,737±0.61% | 2,801±0.001% 3,007±0.22% 2,253±0.29% 1,938±0.34% | 3,003±0.24% 3,001±0.39% 2,249±0.19% 1,746±0.91% | 4,000±0.001% 0,080±0.76% 3,839±0.02% 0,081±0.24% | 7,986±0.14% 0,002±78% 0,012±79% 0,001±78% | |

The results of simulations of M/M/1 and D/D/1 models and the results of our proposed model are shown in Tab.2. and 3.

We made many simulations under different parameter settings. The first 6 presented results correspond with the examples described in the previous section. The simulations 7 and 8 present the difference in the algorithm behavior when changing the priority queue limit to a lower limit than the requirements of the queue and with no

limit. The examples 9 – 20 are always presenting a simulation with a limit set and the same settings with strict priority.

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 [14]. The simulation runs lasted 300 s. The results of the mathematical model mostly correspond with the simulation results. The small inaccuracies can be

TABLE III
SIMULATION RESULTS II.

| Simulation variant [#] | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | |
|---------------------------------|---|--|--|---|--|---|--|---|---|---|------------------------------|
| Mean packet size [B] | 100 1000 1000 1500 | 100 1000 1000 1500 | 100 1000 1000 1500 | 100 1000 1000 1500 | 1000 1000 1000 1000 | 1000 1000 1000 1000 | 1000 1000 1000 1000 | 1000 100 10 1 | 1000 100 10 1 | 1000 1000 1000 1000 | 1000 1000 1000 1000 |
| Mean arrival rate [packets/s] | 100 10 10 1 | 100 10 10 1 | 100 10 10 1 | 100 10 10 1 | 1 10 100 1000 | 1 10 100 1000 | 100 100 100 100 | 100 100 100 100 | 100 100 100 100 | 100 100 100 100 | 100 100 100 100 |
| Weight settings | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 3, 2, 1 | - , 1, 1, 1 | - , 1, 1, 1 | - , 3, 2, 1 | - , 3, 2, 1 | |
| Link capacity [Mbps] | 100 | 100 | 0.05 | 0.05 | 4 | 4 | 0.5 | 0.5 | 1.6 | 1.6 | |
| Priority queue limit [Mbps] | 0.1 | - | 0.03 | - | 0.1 | - | 0.3 | - | 0.7 | - | |
| Model results [Mbps] | 0,08 0,08 0,08 0,012 | 0,08 0,08 0,08 0,012 | 0,03 0,009 0,006 0,005 | 0,05 0 0 0 | 0,008 0,08 0,8 3,112 | 0,008 0,08 0,8 3,112 | 0,3 0,08 0,008 0,001 | 0,5 0 0 0 | 0,7 0,45 0,3 0,15 | 0,8 0,4 0,266 0,133 | |
| D/D/1 simulation results [Mbps] | 0,080 0,080 0,080 0,012 | 0,080 0,080 0,080 0,012 | 0,027 0,011 0,007 0,005 | 0,050 0,000 0,000 0,000 | 0,008 0,080 0,800 3,112 | 0,008 0,080 0,800 3,112 | 0,302 0,080 0,008 0,001 | 0,500 0,000 0,000 0,000 | 0,700 0,450 0,300 0,150 | 0,798 0,401 0,267 0,134 | |
| M/M/1 simulation results [Mbps] | 0,080±1.12% 0,080±2.46% 0,080±3.24% 0,012±10.31% | 0,080±0.69% 0,081±2.34% 0,080±3.05% 0,012±6.93% | 0,025±0.58% 0,011±2.88% 0,008±4.27% 0,006±7.74% | 0,050±0.01% 0,000±0% 0,000±0% 0,000±0% | 0,008±8.54% 0,080±2.96% 0,802±0.84% 3,111±0.26% | 0,008±5.64% 0,080±2.4% 0,797±0.68% 3,114±0.16% | 0,304±0.03% 0,080±0.64% 0,008±0.69% 0,001±0.71% | 0,500±0.01% 0,000±0% 0,000±0% 0,000±0% | 0,701±0.001% 0,450±0.41% 0,301±0.81% 0,149±1.66% | 0,800±0.85% 0,401±1.08% 0,265±1.1% 0,133±1.42% | |

caused by measurement errors, where the bandwidth calculation is stopped closely to an arrival of a packet when small arrival rates are set. For the D/D/1 model, 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 results correspondent with the presented results and proved that this model modification is applicable for both WRRPQ with rate limitation and also strict priority.

7 Conclusion

We presented a new modification of the iterative bandwidth allocation model for WRRPQ in IP based NGN networks. The proposed model uses the weight settings of the scheduler and average input packet size of different flows for the bandwidth calculation. It also uses the rate-limit of the priority queue if set or uses strict priority to assign the bandwidth to the first queue. The variable utilization of different queues and packet redistribution is considered in the calculations. The proposed model allows 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 six different examples and confirmed by simulations in the NS2 simulator for both D/D/1 and M/M/1 input traffics.

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

8 Acknowledgment

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 VEGA No. 1/0106/11 "Analysis and proposal for advanced optical access networks in the NGN converged infrastructure utilizing fixed transmission media for supporting multimedia services" and „Support of Center of Excellence for SMART Technologies, Systems and Services II., ITMS 26240120029, co-funded by the ERDF“.

References:

- [1] E. Chromý, M. Jadron and T. Behul, Admission Control Methods in IP Networks, *Advances in Multimedia*, vol. 2013, Article ID 918930, 7 pages, 2013, ISSN: 1687-5680 (Print), ISSN: 1687-5699 (Online), doi:10.1155/2013/918930.
- [2] E. Chromý, J. Diežka, M. Kavacký, M. Vozňák, Markov Models and Their Use for Calculations of Important Traffic Parameters of Contact Center, *WSEAS Transactions on Communications*, Issue 11, Volume 10, pp. 341-350, November 2011, ISSN: 1109-2742.
- [3] M. Katevenis, S. Sidiropoulos and C. Courcoubetis, Weighted round-robin cell multiplexing in a general-purpose ATM switch chip, *IEEE Journal on Selected Areas in Communications*, vol.9, no.8, pp.1265-1279.
- [4] H. M. Chaskar and U. Madhow, Fair scheduling with tunable latency: a round-robin approach, *IEEE/ACM Trans. Netw.* 11.
- [5] J. Mao, W.M Moh, and B. Wei, PQWRR scheduling algorithm in supporting of DiffServ, *Communications*, 2001. ICC 2001. IEEE International Conference on , vol.3, no., pp.679-684 vol.3, 2001
- [6] Ch. Semeria, Supporting differentiated service classes: Queue scheduling principles, *Juniper Networks*, <http://www.juniper.net>. Part number: 200020-001 12/01
- [7] K. Molnár, M. Vlček, Evaluation of Bandwidth Constraint models for MPLS networks, *Electronics*, 2009, vol. 1, pp. 172-175. ISSN: 1313- 1842.
- [8] RFC 4125. Maximum allocation bandwidth constraints model for DiffServ-aware MPLS traffic engineering. The Internet Society, 2005.
- [9] RFC 4126. Max Allocation with Reservation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering & Performance Comparisons. The Internet Society, 2005.
- [10] RFC 4127. Russian Dolls bandwidth constraints model for DiffServ-aware MPLS traffic engineering. The Internet Society, 2005.
- [11] T. Balogh, M. Medvecký, Mean Bandwidth Allocation Model of WRR for IP Networks, *Telecommunications and Signal Processing TSP-2012 : 35th International Conference on Telecommunications and Signal Processing*, Prague, Czech Republic, July 3-4, 2012. - Brno : VUT v Brně, 2012. - ISBN 978-1-4673-1118-2. - S. 156-160
- [12] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>
- [13] Sergio Androozzi: DiffServ4NS, <http://sergioandroozzi.com/research/network/diffserv4ns/>
- [14] K. Pawlikowski, H.-D. J. Jeong, J.-S. R. Lee, On credibility of simulation studies of telecommunication networks, *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 132-139, Jan. 2002