Simulation Studies of a Multi-priority Dual Queue (MPDQ) with Preemptive and Non-preemptive Scheduling

Anthony Bedford and Panlop Zeepongsekul

Department of Mathematics and Statistics, RMIT University
Plenty Rd., Bundoora East, Victoria, 3083, Australia
{anthony.bedford, panlopz}@ems.rmit.edu.au

Abstract. Traffic management in communication environments like the Internet is vital in meeting the demand of users. The recent introduction of various dynamic scheduling algorithms has been to target improvements in QoS (Quality of Service). These new algorithms, such as dual queueing, aim to give better QoS to most traffic at the expense of a few rather than fairly give poor QoS to all customers. This is achieved through scheduling decisions that can change depending on the traffic conditions. The MPDQ introduces different classes into this type of scheme with the aim of enhancing higher-class packets QoS without great expense to lower class packets. In this paper we analyse the differences in loss and waiting time between preemptive and non-preemptive service disciplines for the MPDQ and then identify the best queueing regime for the pre-emptive MPDQ. We show that non-preemptive service dispensing is superior to packets of two classes than the preemptive scheme, and that highest class first (HCF) is the best queueing regime for either of these algorithms.

1 Introduction

The Multi-Priority Dual Queue (MPDQ) was designed to reduce congestion levels in communications models with finite buffers. By splitting a single finite buffer space into two queues, known as dual queues, and administering a priority scheme to arriving packets, we have shown distinct advantages to communications providers in Quality of Service (QoS) requirements [1]. The MPDQ was derived from the Dual Queue proposed in [3] and combined to include the prioritised packet approach proposed to relieve congestion in [5]. Their analysis in [3] involved the use of MPEG traces used also in [7], which consisted of 20 files. They analysed the delay characteristics of the Dual Queue against the pre-existing scheduling regimes FIFO (First in First Out) and a modified Deficit Round Robin technique (DRR) [8], and demonstrated distinct advantages using the Dual Queueing scheme. This work was extended to a wireless local area network where minor modifications were made to the Dual Queue (DQ) and it was shown to outperform standard Round Robin scheduling in this application for untethered environments [6]. These analyses were undertaken using a fixed amount of

traffic with no statistical assumptions made on the underlying distributions of the arrival or service processes. A simulation approach to Wireless Packet Discard Policies was undertaken in [9], and a simulation approach was undertaken to analyse the scheme in a network scenario. They looked at transition probabilities for the error model and a specific type of loss called Message Loss Ratio. Our contribution explores a multi-priority queueing system that is especially relevant in the Internet Engineering Task Force (IETF) Integrated Services Processes or Differentiated Services architectures. Our modified version of the DQ scheduling discipline provides a simple and effective mechanism for scheduling in these architectures. By introducing different classes into this scheme, we add additional complexity to the DQ with the aim of class-based QoS. Class-based packets within this system of greater importance (or higher class) have precedence over any lower classed packet with the aim of reduced waiting times. This MPDQ differs to the original DQ in that [3] used large finite buffers. As [3] deals with the handling of traffic streams, we change the scheduling to cope with individual prioritised packets, and instigate an abatement of both queues. The setting of the abatement threshold to be equal the capacity of the queues significantly changes the operation of the queues. This is as [3] deals with traffic streams and this paper deals with individual prioritised packets. The MPDQ is especially relevant for the Internet's development, and also has merits for use in certain manufacturing systems where a second holding area is available. In our model, loss is determined by available queue size, and we have not used a time-discard policy, in which packets are ejected from the system after a period of time is reached as in [3]. Through simulation we have shown the benefits of the non-preemptive MPDQ using a Highest Class First (HCF) queueing regime over other queueing regimes [1]. Furthermore, we have developed a detailed algorithmic procedure to explicitly solve the steady state distribution of the preemptive MPDQ for two classes of packets [2]. This technique required the matrix-geometric approach to define the state rate transitions and uniquely implemented them into an algorithm to obtain a solution. That work also highlighted the enormous task of analytically solving even the smallest MPDQ system.

Our aims in this paper are to firstly establish the accuracy of simulation by comparing them with known analytical results. We will then use simulation to explore the differences between the non-preemptive and preemptive service disciplines through extensive simulations using Arena, and explore in great detail the merits of different queueing regimes for the preemptive MPDQ. The results here provide a framework for communications providers in determining how service disciplines contribute to loss and delay, and how modelling can assist in determining how to relieve a network of this type from congestion. In Sect. 2, we define the MPDQ and the simulation design in detail, stating the scheduling algorithms used in the simulations. To investigate the accuracy and reliability of the simulations, in Sect. 3 we compare our analytical results for four models with the simulation results and describe the process of determining a suitable stopping time. In Sect. 4, the comparison of preemptive and non-preemptive MPDQ is undertaken by looking at the results of loss by queue size, waiting time in the system, and waiting time by queue size. This provides valuable insight into the behaviours of the two types of MPDQ's that can be used by service providers. In a similar analysis to our prior simulation studies we complete our investigations in Sect. 5 by looking at the preemptive MPDQ under various queueing regimes. Performance statistics and CDF of delay for four models are fitted, and the waiting time distribution is fitted for Class 1 packets under the HCF MPDQ.

2 MPDQ Design and Simulation

The MPDQ is designed with two finite buffer spaces and a server at the head of the first, or primary, queue. There is a single server (node) at the head of this queue dispensing an exponentially distributed service time with rate μ_i to traffic of class i , $i=1,...,k$ for k classes. If an arriving packet finds the primary queue full, they wait in the secondary queue (provided it isn't full) which has no service facility and only serves as a holding queue for the full primary queue. If both queues are full, an arriving packet is lost. In both queues, packets arrange themselves in order of priority, dependent upon the queueing regime. Here we assume class 1 packets (the high class) have priority ahead of class 2 packets (the low class). The queueing regime used here is, unless otherwise stated, Highest Class First (HCF). The arrivals are independent exponentially distributed arrival processes with mean arrival rate λ_k . Once a space is vacant in the primary queue, the head of the line in the secondary queue joins the primary queue. A schematic diagram describing the MPDQ is given in [1]. In Sect. 5, the queueing regimes investigated are HCF, LCF (Lowest Class First), FIFO, and LIFO (Last in First Out). The regimes stated hold for both queues in the system, so we cannot have a primary queue following a different regime to the secondary queue. For the priority schemes, packets of the same class are sorted on a FIFO basis. The non-preemptive case assumes no interruption to the packet in service, whereas the preemptive service discipline will interrupt service of a packet of lower class for one of a higher class. The lower class packet is returned to the head of the queue and awaits for the server to again become free. They resume requiring full service time.

2.1 Scheduling Algorithm

The simulations were conducted using the Arena Simulation Software package [4], a flexible program suitable for a vast area of statistical analysis with dedicated queueing subroutines. Arena has provided effective results of queueing disciplines in our prior work and was again chosen here as our tool of analysis. Arena allows us to define effective class based simulation due to its ability to generate arriving data and serve data from any statistical distribution. Further we can preempt packets from service if needed, and Arena gathers vast performance characteristics. In constructing the Arena model, we first needed to define the MPDQ in two independent algorithms (for the preemptive and non-preemptive cases). The preemption process required a separate procedure to check for the quantity of first class packets in the system and the preempt subroutine to expel a lower classed packet from service if a higher classed packet is in the primary queue. Both algorithms are simple to extend to multiple classes of packets. The simulation model developed for the non-preemptive simulation was substantially simpler than the preemptive model. Analytically the reverse holds, with the non-preemptive solution being more difficult to obtain than the preemptive solution using the matrix-analytic computational algorithm solution process.

3 Preemptive MPDQ

In [4], we have shown, using complex algorithms, that a steady-state solution for the dual queue with preemptive scheduling is possible. However, the process established is difficult to numerically evaluate due to division of large numbers for even small queueing systems. As the waiting space increases, the states in the system increase rapidly. This can be shown as follows. The steady state distribution π^t exists and is obtained by solving $\pi^t A$ where A is the infinitesimal generator matrix, and 0 is a vector of zeroes. The dimension of A is given by

$$\left(\frac{1}{2}\right)(c_1 + 1)(c_2^2 + 3c_2 + c_1 + 2) \times \left(\frac{1}{2}\right)(c_1 + 1)(c_2^2 + 3c_2 + c_1 + 2) \quad (1)$$

where c_i is the capacity of queue i . So if queue 1 is of size 4 and queue 2 is of size 6, A will have a dimension of 150. A matrix of this size will have 22,500 cells, well beyond a simple matrix multiplication, so simulation provides us with a valuable estimation tool.

The analytical results are a precise solution, based on the long-term steady-state Markovian birth and death processes. Therefore the choice of length for the simulations is vital in attempting to replicate a long-term steady-state result. From the general steady state solution of a birth death process, letting $t \rightarrow \infty$, $\Delta t \rightarrow 0$ solves the time invariant steady state probabilities in differential quotient form. In Arena simulations, there are three possible ways to cease the model running. The first is to set a finite simulation time, the second to define a statistic-dependent conditional statement, and the third is user interruption. Otherwise the simulation will run indefinitely. We use estimated average loss as the statistic-dependent conditional statement to determine a suitable finite simulation time.

3.1 Simulation and Analytical Results

For each of the models, we began with pilot simulations where the proportion of traffic lost is recorded. As expected this approached a constant value as the simulation time increased and will be explored in detail in the next section. We will begin by comparing analytical and simulation results for the probability of a busy server, an empty second queue and of loss. Due to the difficulty in initially finding these results analytically, we only consider queue 1 and 2 sizes of 2. We vary the inter-arrival rates and fix the service rates at 1 sec for Class 1 and 2 for Class 2.

We fixed the service rates in all simulations to the same values, with the details of four models (I–IV) to be used in all simulations given in Table 1. So using the rates used in our previous work [2], we used:

Table 1. Arrival and service rates

Model	Arrivals		Service		Load
	λ_1	λ_2	μ_1	μ_2	
I	1	2	1	2	2
II	.8	1.6	1	2	1.6
III	.3	.6	1	2	.6
IV	.1	.2	1	2	.2

Our first interest was to compare the approximate probabilities for a busy server, π_B , an empty secondary queue, $\pi_{\{Q_2=0\}}$, and loss, π_{Loss} in the simulations with our analytical results obtained in [2]. Our initial simulations were undertaken with the aim of obtaining close results to the precise analytical solutions.

Table 2. Comparison of steady-state and simulation statistics

Model	π_B		$\pi_{\{Q_2=0\}}$		π_{Loss}	
	Steady S.	Simul.	Steady S.	Simul.	Steady S.	Simul.
I	.9539	.9790	.2505	.2454	.5071	.5089
II	.9207	.9556	.3312	.3544	.4421	.4085
III	.5844	.5714	.7659	.8971	.1373	.0437
IV	.2079	.2084	.9742	.9977	.0160	.0001

As seen in Table 2, the results are close, with most probabilities within 0.03 of the actual result. This simulation result provides us with confidence in the reliability and validity of simulation to provide us with an accurate estimator of systems that are beyond current computational power.

To decide on an appropriate simulation time for the later analysis, a cumulative estimate of the mean loss was used. The continuous average probability of loss was calculated for both classes of packet. It is the class 1 loss that was used for the stopping rule as we have defined $\lambda_1 < \lambda_2$. The reason for this is that we consider class 1 packets to be rarer arriving traffic that is of higher importance than class 2 packets. Further as the arrival rate is less, it is expected that the class 2 value should converge quicker to a steady-state. As the simulations were set up to accumulate extensive statistics, the aim was to run an efficient simulation that did not over-gather vast files of needless extra data. Since the traffic loss policy (Q 2 drop tail) is impartial to each class of traffic, it is expected that the probability of loss for both traffic classes should be the same. The loss rates will be proportional to the arrival rates of each traffic class. The pilot simulations for one replication were re-executed using a simulation time of 15,000 units. The generated average loss file for class 1 packet was 3Mb. It was projected that an average loss file over a simulation time of 50,000 units for 10 replications would be approximately 100Mb. The real-time execution period is sensitive to the system load, with the higher the load value, the faster the execution period. The stopping rule was based on the following, where the subscript t is the recorded simulation time as created by Arena, $\hat{\pi}_{Loss1}$ is the approximate probability of loss for a class 1 packet:

$$\left| (\hat{\pi}_{Loss1})_t - (\hat{\pi}_{Loss1})_{t-1} \right| \leq 0.00005 \quad (2)$$

Arena records statistics only if a change in value occurs. This avoids enormous files consisting of non-varying data being recorded. This assists the determination of a suitable simulation time, as the value t is only recorded to file if any of the statistics required have changed. It can be seen that the loss function settles down to a constant value by around 10,000 time units. Using the criteria (2) for stopping, the simulation times for each model were 9753.7, 10199.6, 10041.4 and 3917.5 for Models I, II, III, and IV. From these results a simulation length of 15,000 time units was used for all subsequent simulations. With the simulation design established, we begin our analysis.

4 Comparison of Scheduling Disciplines

The difference between preemption and non-preemption is simple enough to implement for people-type queueing problems but substantial in terms of algorithm design and analytical calculation. Let us define more formally the case where a packet of class i is always taken over a packet of class j . Consider a packet of class j being in service when a packet of class i arrives. Preemption is when the service of the j^{th} class packet is interrupted and the server starts serving the i^{th} class packet, based on a FCFS approach within classes. In non-preemption, when the service of the j^{th} class packet is completed the server starts serving the i^{th} class packet, based on a FCFS approach within classes. By this definition, we would expect Class 1 packets to be advantaged (in terms of expected waiting time) under the preemptive regime over Class 1 in non-preemptive and Class 2 packets under the non-preemptive regime over preemptive regime.

4.1 Loss by Queue Size

In this section, we analyse the effects on loss when we increase the sizes of the queues for each service regime. To undertake this, we use the models stated in Table 1. Since the traffic loss policy is impartial to each class of traffic, it was expected that the probability of loss for both traffic classes should be the same, and that was the case here. The loss rates will be proportional to the arrival rates of each traffic class, and we found that for most queue sizes, class 2 packet loss levels were marginally higher than class 1 loss levels of the same regime. The load gives us a clue as to why we found Model III and IV loss levels approaching zero. From Table 1, we have for Models III and IV $\rho < 1$. Model IV has the smallest load and no loss. These two models shall see arriving packets find a system always with at least one vacant waiting space of the waiting room is extended sufficiently.

What is noticeable from all the models is the marginal improvement in loss levels as queue size increases. In packet based networks, if there are high volumes of traffic, a small decrease in loss levels would lead to a higher volume of packets receiving some service. For most elementary queueing models where $\rho > 1$, the queue length will grow to infinity. In a finite buffer space, loss shall be experienced, and this will only decrease as queue size $\rightarrow \infty$. For packet traffic that follows the distributions stated here, loss levels slowly reduce when addition to the size of the queues is made. As an example, for Model I, in Table 3 below, loss for both classes reduces less than 1% for class 1 packets and less than 1.5% for class 2 packets for a ten-fold increase in queue size.

Table 3. Model I loss by queue qize

Total Queue Size	{Loss1}	{Loss2}
10	.4938	.4986
40	.4836	.4970
100	.4899	.4858

4.2 CDF of Delay

We use the CDF of delay time, or waiting time in the system W_s , to gain a picture of the overall QoS to packets for the two types of service regimes. Each model is investigated, with the expectation of shorter waiting times for class 2 in non-preemptive than preemptive and shorter waiting times for class 1 in preemptive than non-preemptive. In the four Figs. 1a-d, the CDF's follow the expected exponential curve.

In the single queue preemptive and non-preemptive infinite priority queueing models it can be derived from fundamental results that the expected waiting time in the queue for a class 1 packet is less in the preemptive priority case, and it is greater for class 2 packets in the preemptive case. (This is under the condition that $\rho < 1$). From the CDF's for all models, the non-preemptive Class 1 packets outperform all other packets, however the advantage is marginal over the preemptive packets. Clearly, Class 2 packets wait longer as expected than Class 1 packets. For Models I and III, we found an increased chance of shorter waiting times towards the Class 2 over Class 1 packets (for Models I and III, $P(W_s^2 > 2) > P(W_s^1 > 2)$ for both priority schemes). For the Class 2 packets, the preemptive regime is the worst in all models but Model II. For communications providers, the MPDQ for two classes of packets, where $\rho \leq 1$ we can expect Class 1 packets to receive superior waiting time over Class 2 packets. The regime to be implemented using the CDF's is the non-preemptive service regime. It is marginally better for each of the classes than its interrupt counterpart.

Fig. 1a–d. CDF of W_s for models I–IV

4.3 Waiting Time by Queue Size

What is interesting in our analysis is the link between buffer size and waiting time for class 2 packets. From the simulations undertaken here, in models where $\pi\{\text{Loss}\} \rightarrow 0$ both waiting times in the system reach constant values. Furthermore in models where loss is non-zero, waiting time in the system for class 2 traffic increases linearly as queue size increases, and loss for class 1 packets approaches a constant. As nearly all arriving packets will enter the system, a constant waiting time can be achieved for class 1 packets due to their class-based queue jumping. The preemptive class 2 packets is almost identical to the non-preemptive with slight deviations from a straight line the only difference. When we fitted a regression line for class 2 non-preemptive packets we achieved perfect positive correlation, indicating waiting time for class 2 packets is related linearly to queue size.

5 Regimes for Preemptive MPDQ

We now focus on the preemptive MPDQ with the expectation a priority regime is superior to the non-priority queueing regimes. In [1], we found the HCF regime in a non-preemptive model to provide superior QoS to packets over other priority/non-priority regimes. We will describe as in our prior work the waiting time in the system.

Table 4. Mean waiting time in the system by class and queue regime ($c_1=c_2=2$)

Regime	W_s^{class}	I	II	III	IV
HCF	1	2.28	2.20	1.57	1.17
	2	3.04	2.77	1.39	0.74
FIFO	1	3.10	2.96	1.77	1.23
	2	2.59	2.44	1.29	0.72
LCF	1	4.59	4.13	1.94	1.26
	2	1.87	1.77	1.09	0.69
LIFO	1	2.98	2.84	1.73	1.21
	2	2.60	2.38	1.22	0.69

From Table 4, for all of the models the waiting time in the system for Class 1 packets is lowest under the HCF regime. For Class 2 packets, the LCF regime is the best of the regimes. This further justifies the implementation of a priority scheme over a non-priority scheme. The margin between the times decreases as the model number decreases. It would appear that HCF again provides sufficient advantages over the non-priority schemes to warrant its use. Table 4 is the most important for packets as long waiting time is the cause of latency and jitter in many real-time web-based applications. An overall decrease in the difference in the statistics is noticeable as the Model number increases. This should come as no surprise as becomes smaller. So the smaller differences therefore should not be discounted as the traffic volume is high. In Model I, the Class 1 and 2 arrival rates are both one-tenth that of Model IV. Hence the traffic intensity in Model IV is forty times that of Model I. So a small difference in waiting may still lead to a large quantity of packets lost.

5.1 CDF of Class 1 Waiting Times

The CDF of waiting time (system) for Class 1 packets is now investigated for pre-emptive packets. In the models we analysed in the non-preemptive case [1], all of CDF's were superior under the HCF regime. Here we also have the same result. In Figures 7-10, the probability of waiting time in the system by waiting times is given. As ρ decreases, the difference between the curves decreases. The next best waits occur for LIFO, then FIFO and the poorest is LCF.

Fig. 2a-d. CDF of Models I-IV by Regime

It may be that service providers determine the difference is marginal and the priority system too complicated to introduce such a scheme over traditional FIFO. It has been shown in our prior work that the dual queueing system, irrespective of queueing regime, provides superior service over a single queueing scheme for non-preemptive.

6 Conclusion

Through analysis of loss and delay CDF's, we found the best service regime to be the non-preemptive regime. We looked further at the preemptive case and found the HCF the best regime, however as the traffic volume increased, the advantages over the non-priority schemes reduce. For many communications applications, a preemptive service regime is harder to implement as it may require more time to interrupt a packet in service, return it to the head-of-the-line, and recommence service for the higher classed packets than to just let it be served. Certainly, we found the differences small in many cases between preemption and non-preemption. We have provided a detailed framework with which analysts can apply a simulation to analyse a MPDQ system, and results that give tremendous insight into both service regimes for the MPDQ.

References

1. Bedford, A., Zeephongsekul, P.: Simulation Studies on the Performance Characteristics of Multi Priority Dual Queue (MPDQ) with Finite Waiting Room and Non-preemptive Scheduling. In: Kluev, V., Mastorakis, N. (eds.): *Topics in Applied and Theoretical Mathematics and Computer Science*. WSEAS Press, Greece (2001) 220–231
2. Bedford, A., Zeephongsekul, P.: Finite buffer queueing system with preemptive priority scheduling. Research Report No. 9, Dept. Mathematics and Statistics, RMIT University (2002)
3. Hayes, D., Rumsewicz, M., Andrew, L.: Quality of Service Driven Packet Scheduling Disciplines for Real-time Applications: Looking Beyond Fairness. *Proc. IEEE Infocom* (1999) 405–412
4. Kelton, W., Sadowski, R., Sadowski, D.: *Simulation with Arena*. 2nd edn. McGraw-Hill, (2002)
5. Odlyzko, A.: The Current State and Likely Evolution of the Internet. *Proc. IEEE Globecom* (1999) 1869–1875
6. Ranasinghe, R., Andrew, L., Hayes, D., Everitt, D.: Scheduling Disciplines for Multimedia WLANs: Embedded Round Robin and Wireless Dual Queue. *Proc. IEEE Int. Conf. Commun.* (2001) 1243–1248
7. Rose, O.: Statistical Properties of MPEG Video Traffic and their Impact on Traffic Modeling in ATM Systems. *Proc. 20th Ann. Conf. on Local Computer Networks* (1995) 397–406
8. Shreedhar, M., Varghese, G.: Efficient Fair Queuing using Deficit Round Robin. *IEEE/ACM Trans. Networking*, Vol. 4, No. 3 (1996) 375–385
9. Siripongwutikorn, P., Labrador, M., Znati, T.: A Wireless-aware Packet Dropping Policy for ATM Networks. *Proc. Comm. Networks and Distributed Systems Modeling and Simulation Conf.* (2000) 155–160