The Role of Packet-dropping Mechanisms in QoS Differentiation

Goncalo Quadros, Antonio Alves, Edmundo Monteiro, Fernando Boavida
CISUC – Centro de Informática e Sistemas da Universidade de Coimbra
Departamento de Engenharia Informática
{quadros, aalves, edmundo, boavida}@dei.uc.pt

Abstract

Many research teams are developing technologies to turn the Internet into a QoS-capable network, which is one of the biggest challenges that this communication system currently faces. Naturally, at the core of such a challenge are IP routers and the technology they use.

It is a well known fact that the common packet scheduling discipline that is used in routers (first come first served) makes them useless when QoS is needed. Thus, a different type of packet scheduling must be used. One of the most referred solutions for QoS-capable systems is the Weighted Fair Queuing (WFQ) discipline. For FreeBSD-based routers, the ALTQ implementation of the WFQ discipline is, of course, an eligible and natural choice. Given this, it is important to fully understand the characteristics and operational behaviour of such an implementation.

This paper presents several tests that guide the reader to a detailed knowledge about the WFQ/ALTQ operation – its behaviour, weaknesses, and flaws – with the purpose of showing how relevant can the influence of the dropper mechanism be on the effectiveness of IP routers.

INDEX TERMS -- WFQ/ALTQ, IP packet dropper, IP QoS

1. Introduction

In recent years, work concerning the provision of controlled quality of service (QoS) to different Internet flows has led to numerous studies and proposals of mechanisms to build a new service model for IP networks. Many of these proposals focus on scheduling disciplines to guarantee and/or differentiate the performance given to classes of data flows, and normally rely on dropping strategies for congestion avoidance [Floyd95, Floyd93, Braden98, Keshav91, Bennett96]. Despite the fact that the combined effects of such mechanisms (packet scheduler and dropper) are not always clear, these mechanisms must coexist in any real router. The main intention of this paper is to show the effect of packet dropping mechanisms on QoS provision, discussing some experimental results obtained in the context of a project being carried out at LCT-CISUC[1].

The referred project has a relative broad scope that spans the development of a new approach for supporting traffic classes in IP networks. The idea is to use a multiple-class-best-effort model instead of the current single-class-best-effort Internet paradigm, protecting more sensitive classes and letting less sensitive classes absorb degradation. This requires continuously measuring the quality of service provided to classes and to re-allocate system resources according to a given fairness criterion.

A fundamental step for QoS provision is the selection of an alternative to the common FIFO discipline used in routers. As Intel/FreeBSD platforms were being used at LCT-CISUC for the purpose of the project, it was decided to use the ALTQ technology [Cho98] as an alternative to the traditional IP queuing system. The Weighted Fair Queuing (WFQ) implementation of the ALTQ project (WFQ/ALTQ) was chosen because of its simplicity and because it seemed capable of supporting the dynamic control of the QoS provided to classes – a fundamental requirement of the project – through the adjustment of the classes' weight. To evaluate the effectiveness of WFQ/ALTQ in the support of class differentiation it was decided to submit it to a set of tests.

This paper presents the results of the above-referred tests. Section 2 details the test environment, describing the testbed and the used tools. Section 3 presents the tests, their results and the corresponding analysis. This analysis shows that, in addition to the scheduling discipline, the packet-dropping mechanism can be of major importance in QoS differentiation. Section 4 summarizes the test results and their main conclusions, and positions the presented work in the global LCT-CISUC on-going project for the implementation of a new IP service model.

2. Test environment

The main goal of the tests carried out on the WFQ implementation of the ALTQ project were twofold:

1. to evaluate its real capacity to differentiate traffic;

[1] Laboratory of Communications and Telematics.

2. to evaluate how well, and how easily, it was possible to control the performance provided to different traffic flows.

In the next section the testbed used for supporting this work is presented. Afterwards, the main tools used during the tests are introduced.

2.1 The testbed

Figure 1 details the testbed that was used to evaluate WFQ/ALTQ. It consisted of a small isolated network with 4 Intel Pentium PC machines configured with a Celeron 333MHz CPU, 32 MB RAM and Intel EtherExpress Pro100B network cards.

The PC router - named ROUTER in the figure - ran the operating system FreeBSD version 2.2.6, patched with ALTQ version 1.0.1. It was configured for using the WFQ/ALTQ technology on the interface connected to SINK –hereafter named 'output interface'. The default configuration of WFQ/ALTQ was used, which means that, on average, the scheduler processed a maximum of 512 bytes each time it visited a queue with an assigned weight of 100. If the assigned weigh were 50, the scheduler would process 256 bytes; if the assigned weight were 200, the scheduler would process 1024 bytes, and so on.

The purpose of SOURCE1 and SOURCE2 was to generate traffic destined to SINK. The idea was to generate independent traffic flows, by generating, at each host, traffic of a single and exclusive class.

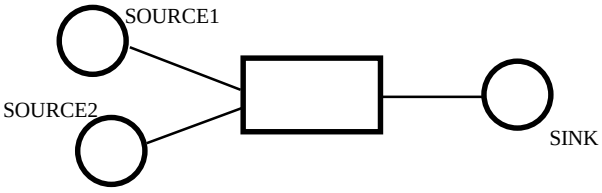


Figure 1 – The Testbed

2.2 The tools

Two different tools were used to perform the tests. *Nttcp* [NTTCP] is a public domain application chosen for generating traffic. As explained below, this tool was used to generate flows of fixed length packets at the maximum possible rate.

The other tool was *QoStat* [Alves99] – a tool implemented at LCT-CISUC and freely available on its site. This tool was nuclear to the work presented in this paper. Through it, it was possible to graphically view, in real time, IP traffic measures related to each of the different traffic classes. For instance, this tool was used to view the number of packets processed and dropped per unit of time, the average packet transit delay at the IP layer, and the average and sampled queue lengths.

In addition, QoStat also allows to change, on the fly, some fundamental parameters related to the operation of the system being monitored. For instance, in this case, it was used to change queue weights and queue lengths. In short, QoStat provided the means to study the exact influence of the different operational parameters on the actual QoS provided by the router. Lastly, it is noteworthy to say that all the graphics presented in this paper were produced by the QoStat tool.

Figure 2 logically presents the system that was submitted to tests. Notice that the dropper mechanism implements a policy that is different from the most common and simple tail-drop policy. The WFQ/ALTQ dropper will be extensively presented below.

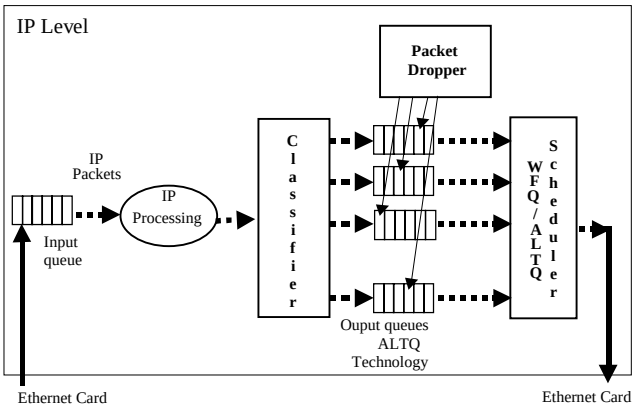


Figure 2 - IP architecture at ROUTER

3. Tests Results and their Analysis

The first set of tests used UDP traffic only and, to facilitate the generation of heavy loads, large packets – an IP payload length equal to 1450 bytes. Under these conditions, and knowing the characteristics of the WFQ discipline, a very good behaviour of the tested router was expected in what concerned its capacity to treat traffic classes according to their weight.

The general test strategy was the following:

1. to use two different and independent traffic classes, composed of IP packets with large, constant lengths, generated by two different hosts at the maximum possible rate;
2. to fix the weight associated with class 1[2] to 100; and
3. to vary the weight associated with class 3[3], successively taking the values of 30, 70, 100, 500 and 1000.

[2] Whose queue is named q1 in all the figures throughout this paper.

[3] Class 2 was not used in the tests. Additionally, as only classes 1 and 3 were used, the measured values for class 0 were always null.

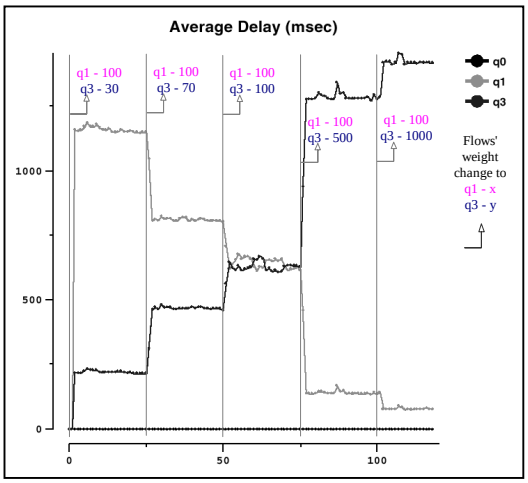
Class 3 weights were changed during the tests' execution at the end of each 25-seconds time interval, using the *QoStat* tool. This tool was also used to monitor the actual performance given by the ROUTER system to the different traffic classes. More precisely, the following values were monitored, on a per class basis:

- average packet transit delay at the IP layer, measured over 1-second time intervals; a packet transit delay is the time that elapses between its enqueuing at the IP input queue, and its dequeuing from the IP output queue (see figure 2);
- number of packets sent through the output interface per 1-second time interval;
- number of dropped packets per 1-second time interval that should have been sent through the output interface;
- maximum IP output queue length obtained in each 1-second time interval.

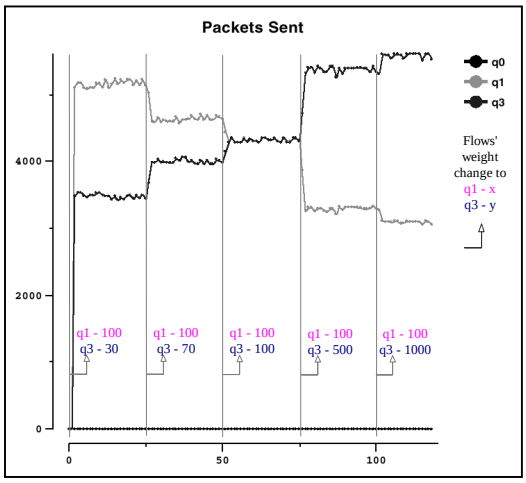
The results of the tests are presented in Figure 3. This figure shows that in the conditions of the tests WFQ/ALTQ is, in fact, able to differentiate traffic. It is also possible to see that there is a clear influence of the weight on the performance attained by the classes.

Nevertheless, one important evidence – coming out also from figure 3 – is that, contrary to all expectations, the packets belonging to the class with highest weight are the ones which suffer highest transit delay in the IP layer. So, at least in what concerns transit delay, the behaviour of the WFQ/ALTQ seems to be totally inadequate.

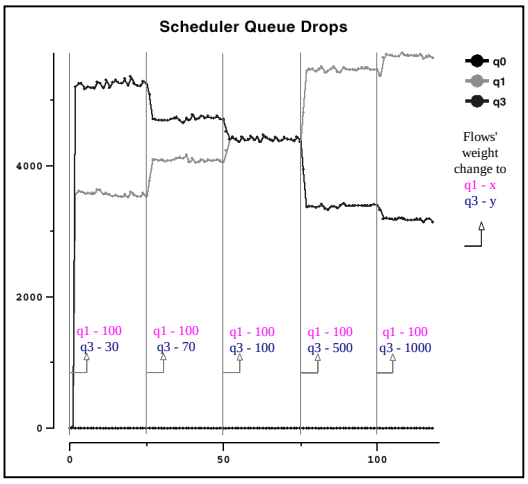
a)



b)



c)



d)

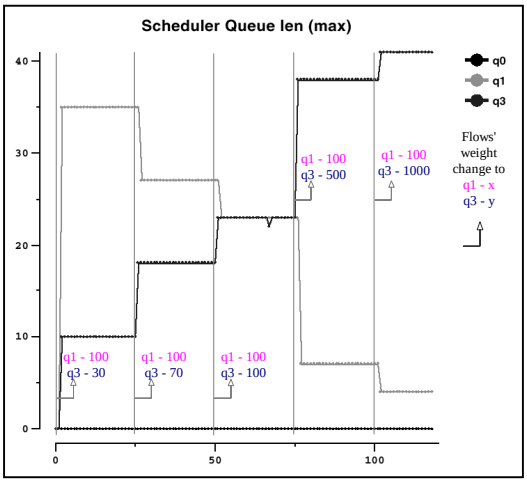


Figure 3 – Tests made to the scheduler with maximum output queue lengths equal to 50 pcks. Variation of the following values with weight: a) Average IP transit delay (over 1second time intervals); b) Number of pcks sent per second; c) Number of dropped pcks per second; d) Instantaneous IP output queue length (sampled every second)

3.1 Analysis of results

The results shown in Figure 3 reveal some important facts. In addition to the referred unexpected behaviour in terms of transit delay, it is also possible to see that there is no proportionality between the class weights and the bandwidth they get (Figure 3b). Given that the tests were carried out using very high loads (twice as much as the router could handle), one would expect that there would be packets in both queues (the same is to say, packets of both classes) at almost all times. In this case, the number of packets sent per unit of time for either class should be proportional to their weight, and the average transit delay per packet should be inversely proportional to those values.

However, that does not correspond to what Figure 3 shows. In order to try to find an explanation for this phenomenon, some specific tests to the ROUTER system were carried out. The QoStat tool was modified in order to systematically count the number of times the scheduler was executed per second and the number of times it found packets in both queues, in one queue only and in no queues at all. Surprisingly, the scheduler found packets to process in both queues only in 21% of cases, despite the fact that UDP traffic and huge loads were being used.

This is due to the system dynamics, namely to the way packets are de-queued from the WFQ/ALTQ buffers. The routine responsible for this task - ifstart() - loops as long as there are packets waiting in the WFQ/ALTQ queues and there is enough space in the Ethernet software buffer. This routine is also called when a new packet arrives at the router. The routine runs at the highest priority level (*splimp*) preventing the kernel code to refill the queues with new packets when the Ethernet driver is being executed [Risso99].

The impact of this behaviour depends on the type of scheduling being used. WFQ theory says that when the scheduler is ready for transmission, at time τ, it selects the packet that would be serviced first if the Generalised Processor Sharing (GPS) were being used. WFQ is a packet approximation algorithm of the GPS discipline, which has some desirable characteristics but, since it uses an idealised fluid model, it cannot be implemented in the real world[4]. WFQ/ALTQ is not a true implementation of the *Weighted Fair Queuing* discipline. It is, in fact, closer to a variant named *Sthocastic Fair Queing* [McKenney90].

Most important for this analysis is the fact that the discipline used in WFQ/ALTQ, as well as in the WFQ and WF2Q disciplines, is a work-conserving discipline. Therefore, it needs packets in the various queues in order to differentiate the performance given to the traffic classes, reflecting their configured weight. When the scheduler finds packets in only one queue it will process the packet at its head immediately. So, if there are two flows (of two different classes) competing for router resources, and if their packets appear alternately at the router queues, the flows will receive exactly the same treatment independently of the weight configured for each class. As this is the case in the major part of the time of the above-mentioned test, the results shown in Figure 3 (namely the WFQ/ALTQ capacity to differentiate traffic) must derive from some reason other than the scheduler operation.

After a thorough analysis of the WFQ/ALTQ implementation, an hypothesis able to justify the behaviour observed in the ROUTER system was reached: the capacity to give different performance levels to classes, according to their weight, stems from the dropper strategy used in the WFQ/ALTQ implementation and not from the operation of the scheduler itself. The remainder of this section shows why this hypothesis was developed and how it was definitely confirmed.

The dropper included in the WFQ/ALTQ implementation works as follows:

1. whenever a packet arrives to one of the IP output queues (see figure 2) the volume of data stored in all those queues is compared with the value of a *high water mark* (which defaults to 64K bytes);
2. if the former value is greater than the latter, the dropper mechanism is called;
3. this mechanism will drop the packet at the head of the queue with the biggest size.

A key issue in the behaviour of the dropper is how it evaluates the length of the queues. In fact, for the dropper the length of the queue is not its real size but the number of bytes in the queue *normalised* by the correspondent class' weight. The dropper uses the following formula to calculate a given queue length:

$$Queue\_length = bytes\_in\_queue * 100/class\_weight \quad (1)$$

The use of such a formula induces an important trend in the system dynamics: queue lengths will tend to follow the weights of the respective classes. Queues of classes with higher weights will most probably have bigger lengths than the queues of the classes with lower weights, and the difference in their sizes will tend to be proportional to the difference in the correspondent weights.

To better understand what has just been mentioned, suppose two traffic classes exclusively composed of IP packets with a fixed length of 1500 bytes were being

[4] WF2Q is presented in [Bennett96] has having better characteristics than WFQ. The only difference is that packets are chosen not from all the queued packets but, instead, from those which would have started receiving service at time τ if the GPS discipline was being used.

used. Additionally, suppose that a weight of 100 was assigned to one of the classes and a weight of 1000 was assigned to the other. Using the default *high water mark* there could only be 44 packets simultaneously stored in all of the output queues (grossly, 64k/1500). Given the dropper behaviour, the length of the queue corresponding to the class with weight 100 would tend to 4 packets as long as the length of the other queue would tend to 40 packets.

In fact, for the dropper the lengths of both queues are equal when they contain the referred number of packets: for the class with weight equal to 100, expression (1) results in the value 5800 (`4*1450*100/100`); for the other class, the formula leads to the same value (`40*1450*100/1000`). In short, in situations of extremely heavy loads, the dropper activity will result in queue lengths that will tend to 4 and 40 packets, respectively.

It is this asymmetry in queue lengths that is responsible for the router behaviour seen in Figure 3. In the following, this statement is explained, starting with the observed number of packets sent per unit of time.

Recall that the experiments revealed that most of the time the scheduler found a single queue with packets. Naturally, the queue with the higher probability to be found non-empty is the one with the higher average length (in other words, the probability to find one non-empty queue will increase with its average length). Thus, the number of packets of a given class processed per unit of time will be proportional to the average length of the corresponding queue. But, as shown above, the queue length generally follows the queue weight. Thus, the number of packets processed per unit time reflects the values of the classes weights – as, in fact, can be observed in Figure 3.

Figure 3 shows no proportionality between the weights of the classes and the number of the respective packets processed per unit of time. Given the last paragraph, this could be seen as an unexpected result. Nevertheless, this can be easily understood if one realises that the weight influences, through the dropper, the length of the queue, but does not determine its exact value nor its average value (which will be highly dependent on the systems dynamics). Thus, it can be expected that changing the classes' weights can result in some variation of equal signal in the number of its processed packets per unit of time, and no more than that. Conversely, it can be expected that changing the classes' weight will produce a variation of opposite signal on the number of packet drops per unit of time.

Given the analysis made so far, explaining the behaviour of the router in terms of transit delay – as observed in Figure 3 – is straightforward. The higher the class weight, the higher the average length of the corresponding queue, and so, the higher the transit delay of its packets.

In short, it can be concluded that the capacity to differentiate traffic shown by the router – well depicted in Figure 3 – is not due to the WFQ scheduler operation. Instead, it is due to the operation of the dropper mechanism, which induces a strong asymmetry of queue lengths.

Despite the strong evidences that were found, it was decided to unequivocally test the above-mentioned arguments. A specific set of tests was designed for this purpose. The WFQ/ALTQ code was changed in order to control the type of dropper policy to use. In addition to the original mechanism, a simple tail-drop mechanism was included. Through a button in the QoStat tool, it was possible for the user to choose which mechanism to use, at any instant in time.

With the tail-drop scheme, the maximum queue length of each class becomes constant and equal to 50 packets. Thus, with high and uniform loads, each queue length will tend to grow to a maximum of 50 packets, independently of its weight. Moreover, the weights will not induce any difference in the lengths of the queues, and so the dropper will not be able to induce any differences in the way packets from different classes are treated.

The tests were a repetition of those whose results are shown in Figure 3. Two traffic flows were generated at the maximum possible rate; one was assigned to class 1 and the other to class 3. The weight of class 1 was fixed at the value 100. The weight of class 3 was successively varied through the values 30, 70, 100, 500 and 1000, after each 25-seconds interval. In this test, however, each of these 25-seconds interval was further split into two parts. In the first part, the original dropper mechanism was used. In the second part, the tail-drop mechanism was used. This was done in order to determine the exact influence of the dropper on the behaviour of the router patched with the WFQ/ALTQ technology. The results of this test are shown in Figure 4.

In this figure, it is possible to see that whenever the original dropper mechanism is used the router differentiates the performance given to the traffic classes, exactly as seen in Figure 3. However, when the mechanism is changed to the tail-drop scheme, the capacity to differentiate traffic disappears completely. In fact, the number of packets sent per unit of time is equal for both classes and does not vary with the change of the classes' weights during the intervals where the tail-drop mechanism is being used. The same is true for the packets' average delay.

Given this, it is possible to unambiguously conclude that the hypothesis previously formulated was correct. The capacity to give different performance levels to different traffic classes, revealed by WFQ/ALTQ in the

carried out tests, results from the dropper operation and not specifically from the WFQ scheduler operation.

This last set of tests also shows the role that packet-dropping mechanisms can have in the operation of any system aimed at QoS differentiation. No matter how well the system behaves under no loss conditions, this behaviour can be totally altered under heavy load if the packet-dropping mechanism is inadequate. These tests highlighted the fact that the packet-dropping policy is an essential part of any technique that aims at QoS provision, along with the scheduling policy, and can contribute in a decisive way to a controlled and QoS-capable behaviour of routers, under any load conditions.

As a consequence, the proposal and development of QoS-capable routers that integrate scheduling and dropping mechanisms in an effective way seems to be the next logical step. This is presently being pursued at LCT-CISUC.

4. Conclusion and Future Work

At the Laboratory of Communications and Telematics of the University of Coimbra the authors have been working on a new IP service model [Quadros99a]. One important piece of any service model is the scheduler. Because of its simplicity, availability and openness, it was assumed, at first, that the ALTQ implementation of the Weighted Fair Queuing would be an interesting choice. This paper presented some of the tests that were carried out on this scheduler.

The tests revealed some important flaws of the WFQ/ALTQ behaviour, concerning its capacity to differentiate the performance given to different traffic classes. The most important one was its inability to consistently control the transit delay given to classes. Surprisingly, it was found that higher classes' weights correspond to worse packet transit delays.

In the attempt to understand such an unexpected behaviour, further tests showed that the WFQ/ALTQ behaviour related to QoS provision stems from the operation of the used dropper mechanism, which induces an asymmetry in the length of the classes' queues, and not from the WFQ discipline implementation. The work-conserving nature of the WFQ discipline is a killing characteristic when the system dynamics results in difficulties to guarantee the simultaneous presence of packets in more than one queue, which is the case in INTEL/FreeBSD platforms.

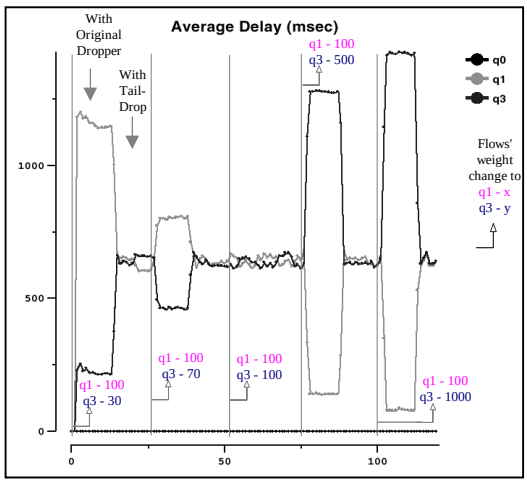
The main contribution of the present paper is the demonstration of the importance of carefully designing the integrated operation of scheduler and dropper mechanisms, when developing routers able to provide quality of service. Without an integrated design, it is highly probable that unexpected and undesirable results will be obtained.

As future work, the authors plan to use the lessons learned with the experiments presented in this paper and in [Quadros99c] to conceive a QoS-capable router prototype, to be used with the IP service model already under development at LCT-CISUC.

Acknowledgment

This work was partially supported by the Portuguese Ministry of Science and Technology (MCT), under the program Praxis XXI - PRAXIS/P/EEI/10168/1998 (Project QoS II - Quality of Service in Computer Communication Systems).

a)



b)

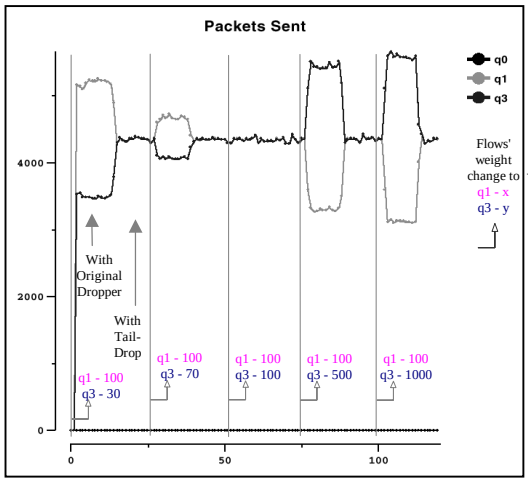


Figure 4 – Tests made to the scheduler with maximum output queue lengths equal to 50 packets. Variation of the following values with weight: a) Average IP transit delay (over 1second time intervals); b) Number of packets sent per second).

References

[Alves99] Antonio Alves, Goncalo Quadros, Edmundo Monteiro, Fernando Boavida, *QoStat – A Tool for the Evaluation of QoS Capable FreeBSD Routers*, Technical Report, CISUC, July 99.

[Bennett96] J.C.R. Bennett and H. Zhang, ``WF2Q: Worst-case Fair Weighted Fair Queueing", INFOCOM'96, Mar, 1996.

[Braden98] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, Re-commendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309, April 98.

[Cho98] Kenjiro Cho, A Framework for Alternate Queueing: Towards Traffic Management by PC Based Routers, in *Proceedings of USENIX 1998 Annual Technical Conference*, New Orleans LA, June 1998.
www.csl.sony.co.jp/person/kjc/kjc/papers/usenix98

[Floyd93] S. Floyd et al., "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993 - http://www-nrg.ee.lbl.gov/floyd/papers.html.

[Floyd95] S. Floyd et al., "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, August 1995 - http://www-nrg.ee.lbl.gov/floyd/papers.html.

[Keshav91] S. Keshav, "On the Efficient Implementation of Fair Queuing", Internetworking: Research and Experience, September 1991.

[McKenney90] P. McKenney, Stochastic Fairness Queueing, in Proceedings of IEEE INFOCOM, San Francisco, California, June 1990

[NTTCP] New TTCP Program.
www.leo.org/~bartel/nttcp/

[Quadros99a] Goncalo Quadros, António Alves, Edmundo Monteiro, Fernando Boavida, “An Approach to Support Traffic Classes in IP Networks”, to be published in Proceedings of QofIS’2000 – The First International Workshop on Quality of future Internet Services, Berlin, Germany, September 25-26, 2000.

[Risso99] Fulvio Risso, Panos Gevros, "Operational and Performance Issues of a CBQ router", Computer Communication Review, Volume 29, Number 5, October 1999.