Simulation-Based Performance Comparison of Queueing Disciplines for Differentiated Services Using OPNET

Article

2 authors:

Hafiz Asif
COMSATS Institute of Information Technology
13 PUBLICATIONS 9 CITATIONS

SEE PROFILE

El-Sayed M. El-Alfy
King Fahd University of Petroleum and Miner…
90 PUBLICATIONS 193 CITATIONS

SEE PROFILE

Simulation-Based Performance Comparison of Queueing Disciplines for Differentiated Services Using OPNET

Hafiz M. Asif and El-Sayed M. El-Alfy
College of Computer Science and Engineering
King Fahd University of Petroleum and Minerals
Dhahran 31261, KSA
Email: {mahafiz, alfy}@kfupm.edu.sa

Abstract

Queueing disciplines have been a subject of intensive discussion and research in the network field for scheduling packets from different traffic flows for processing at a network node (switch or router). Several queue scheduling disciplines have been designed to improve the quality of service for networked multimedia applications. In this paper, we evaluate the performance of four queueing disciplines (FIFO, PQ, WFQ and DWRR) that are widely deployed for differentiated services using a hypothetical network topology. We carry out simulation using OPNET and compare their relative performance based on queueing delay, packet drop rate, end-to-end delay and delay jitter with drop-tail and RED policies.

1. Introduction

As the growth of network applications and internetworking is rising exponentially, numerous requirements are also emerging on the screen. Supporting current and emerging multimedia applications requires the network to ensure acceptable level of service to packet flows over shared and scarce resources (e.g. link bandwidth and buffer space). For example, real-time traffic such as interactive voice can not tolerate delay of more than 250ms. File transfer, on the other hand, is more sensitive to packet loss but can tolerate delay. Compressed video is more sensitive to delay variation or jitter but is flexible in terms of packet loss (up to 1% is acceptable without severe degradation of quality). Several mechanisms have been proposed and deployed in today's network devices to support required QoS such as resource reservation, admission control, traffic shaping, scheduling and packet discarding policies [1]. The cheapest and most common approach is scheduling mechanisms and packet discarding policies. A queue scheduling discipline manages the allocation of network resources among different traffic flows by selecting the next packet to be processed. Also when packets arrive faster than they can be processed, arriving packets are dropped and thus control the congestion in the network. Furthermore, it can reduce the impact of ill-behaved flows on other flows especially when having a mixture of real-time and non-real-time traffic. Two architectures that essentially depend on scheduling techniques have been proposed for supporting QoS in the Internet: *Integrated Services Architecture* (ISA) [2] and *Differentiated Services Architecture* (DSA) [3]. Several variations of queue scheduling disciplines have been proposed in the literature [4, 5]. In this work, we examine four queueing disciplines which are widely deployed and compare their performance using simulation carried out in OPNET [6]. The queueing disciplines considered in this paper are: First-In-First-Out (FIFO), Priority Queueing (PQ), Weighted Fair queueing (WFQ) and Deficit Weighted Round Robin queueing (DWRR). We also scrutinize the effect of two dropping policies, drop-tail and random-early drop (RED) [7], on their performance.

The remainder of the paper is organized as follows. In the next section, we review related work and briefly discuss the operational aspects of the selected queue scheduling disciplines with some comments on their performance. In Section 3, we describe the simulation model and traffic scenarios. We then present the simulation results and evaluate the performance of different queueing disciplines in Section 4. Finally, we conclude our paper and summarize results in Section 5.

2. Related work

We have selected four popular queueing disciplines for our experiments: FIFO, PQ, WFQ and DWRR. In this section, we briefly describe related work and some operational aspects and related performance issues.

First-In-First-Out (FIFO) queueing is the most popular queue scheduling discipline that has been extensively examined in the literature. It also serves as a baseline for comparing the performance of other queue scheduling disciplines. In FIFO queueing, all packets placed into a single queue and then served in the same order on which they arrived. Hence, it is also known as firstcome-first-serve (FCFS) queueing. Although it is simple and has predictable behavior and extremely low computational load on the system, it has some severe limitations for multimedia traffic. It is incapable of providing differentiated service and can not isolate the effect of ill-behaved flow on other flows. A bursty flow can consume the entire buffer space and causes all other flows to be denied service until after the burst is serviced. This can result in increased delay, jitter and loss for the other well-behaved flows traversing the queue. To reduce the impact of ill-behaved sources, other queue disciplines have been proposed to isolate traffic flows into separate queues. These queues can be serviced according to some scheduling scheme. Among these are priority queueing, fair queueing, weighted fair queueing, weighted round robin or class-based queueing, and deficit weighted round robin.

Priority Queueing (PQ) is a simple approach to provide differentiated services to different packet flows. Packets of different flows are assigned a priority level according to their QoS requirements. When packets arrive at the output link, they are first classified into different classes enqueued separately based on their priorities. Then, queues are served in order. The highest priority queue is served first before serving lower priority queues. Packets in the same priority class are serviced in a FIFO manner. As soon as high-priority packets are served, packets from the lower priority class are served. But if a higher-priority packet arrives while serving a lower-priority packet, the server waits until complete the service of the current packet then goes back to serve the higher priority queue (non-preemptive PQ). The limitation of PQ is that lower-priority packets may receive little attention when a higher-priority class has a continuous stream of packets. This problem is known as starvation problem [1]. Also it lacks fairness. The closed form analytical solution of a two-class priority queue is given in [8].

Weighted Fair Queueing (WFQ) [9] is an attractive technique that plays a central role in integrated and differentiated services architectures [1-2]. WFQ was inspired by the work on fair queueing by J. Nagle in 1987 to resolve some of its limitations when mixing several flows of different packet sizes on a shared link. Similar to PQ, arriving packets are first classified and assigned to one of per-class queues based on information taken from the packet header. Each queue is assigned a weight based on the required bandwidth. WFQ has several advantages. It can provide a weighted fair allocation of resources to each service class independent of the behavior of other classes when combined with traffic shapers at the edges of the network. However, it is computationally complex and the supported delay bounds can still be large [4]. Also, it can not handle ill-behaved flows within the same service class. Several attempts to balance complexity and performance of WFQ have been proposed and implemented for differentiated services [10-13].

Deficit Weighted Round Robin (DWRR) was proposed to guarantee QoS with respect to delay and bandwidth for real-time burst traffic [14]. DWRR was proposed by M. Shreedhar and G. Vaghese in 1995 to address the limitations of WFQ and a prior algorithm known as Weighted Round Robin (WRR). Similar to WFQ, packets are first classified into various service classes and assigned to separate queues then serviced in a round-robin order. However, each queue maintains a deficit counter that specifies the total number of bytes that the queue is permitted to transmit. Each time a queue is visited by the scheduler the deficit counter is first increased by a quantum of service proportional to its weight. Then, the scheduler transmits the packets at the head of the queue and decrement the deficit counter by the size of the transmitted packet as long as the size of the packet is not greater than the deficit counter. If the queue becomes empty, the deficit counter is set to zero. Then, the scheduler moves to the next non-empty queue. DWRR has several strengths over other queue disciplines such as lower complexity and better control of network resources especially when serving variable-size packets. For detailed discussion of DWRR, we refer the reader to [4, 14].

3. Simulation model

In order to demonstrate the performance of different queueing disciplines, we considered a hypothetical network topology as illustrated in Fig. 1 in our simulations. There are two routers connected by a DS1 link and all other links are 10BaseT. There are four different traffic scenarios, one for each queue scheme. There are three types of traffic: FTP, Voice and Video. There is a separate server for each traffic type. To evaluate and compare the performance of different queueing disciplines, we collected the average queueing delay, the average end-to-end delay, the average packet drop-rate and the average delay jitter for each queue scheme. We also studied the effect of using random-early drop (RED) and tested its dominance over drop-tail policy.

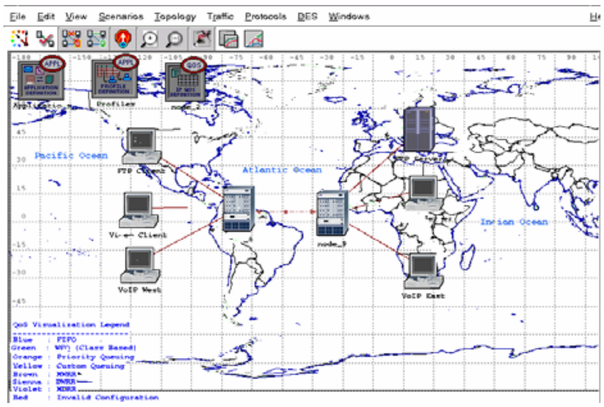

Fig. 1. Network topology for simulation.

For FTP traffic, we used exponential distribution for packet arrival, constant packet size and best-effort type of service. For video traffic, we used low resolution video starting at 10 fps (frames per sec) arrival rate and 128x120 pixels and keep increasing this rate and size as load increases. The ToS is Streaming Multimedia. For voice traffic, the voice encoder scheme is G.711, the silence and talk spurt lengths are exponentially distributed and ToS is Interactive Voice. All these settings were made using OPNET Application Attributes Profile.

For queueing disciplines specification, we set the maximum queue size to be 500 packets. We imply OPNET default weights in case of WFQ and DWRR. There are three types of classes in all queues (except FIFO) whose queue buffer size increases from highest priority to lowest one. Finally when RED is enabled, we used minimum and maximum threshold as 100 and 200 respectively while keeping the mark probability denominator (the fraction of packets dropped when the average queue size is at maximum threshold) as 10. We set exponential weight factor (used to calculate average queue size based on the previous average and current queue size) to be 9.

4. Simulation results and performance comparison

4.1. Queueing delay

Figure 2 shows the average queueing delay of the four queue disciplines (with drop-tail policy) versus simulation time. We can see that FIFO has the worst behavior (more than 2.5 msec) as compared to other disciplines that differentiate traffic.

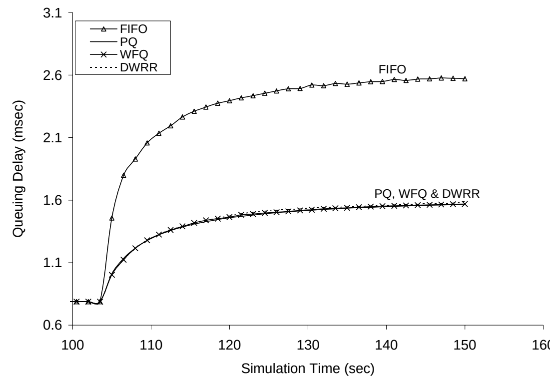

Fig. 2. Queueing delay with drop-tail policy.

Figure 3 is similar to Fig. 2 but on a finer scale of the delay axis to clearly illustrate the relative performance of PQ, WFQ and DWRR.

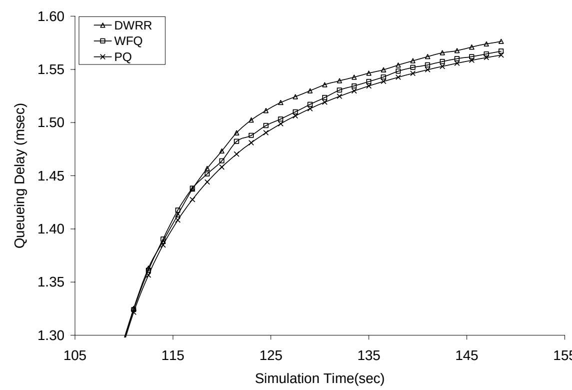

Fig. 3. Queueing delay with drop-tail policy on fine scale.

To study the load effect on the queueing-delay performance, we run the simulator under different loads and plot the queueing delay variation versus load as shown in Fig. 4. We change the load by changing IAT (inter-arrival time) in FTP client, and frame size and rate in video client. We can see that the performance of DWRR supersedes other schemes as the load increases. This result is totally in agreement with what we discussed in Section 2.

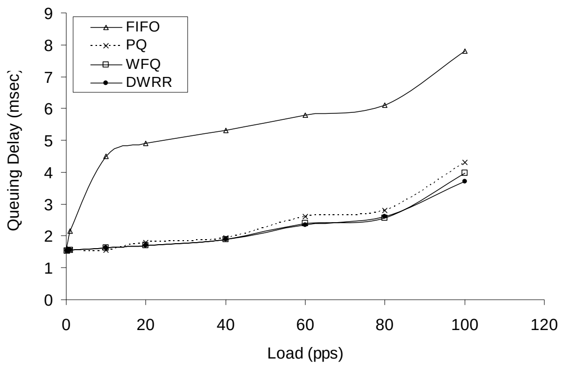

Fig. 4. Queueing delay versus load with drop- tail policy.

The results with RED policy are shown in Fig. 5 (queueing delay vs. simulation time) and Fig. 6 (queueing delay vs. load). As shown, RED has greatly improved the performance of FIFO as compared to the previous results with drop-tail policy. There is no significant improvement for other queueing disciplines in terms of queueing delay but we'll see the effect is clear on delay jitter and end-to-end delay.

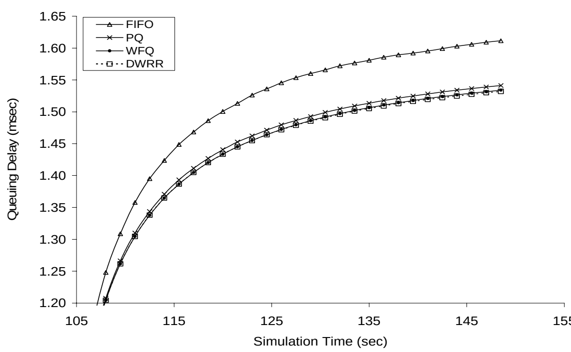

Fig. 5. Queueing delay with RED policy.

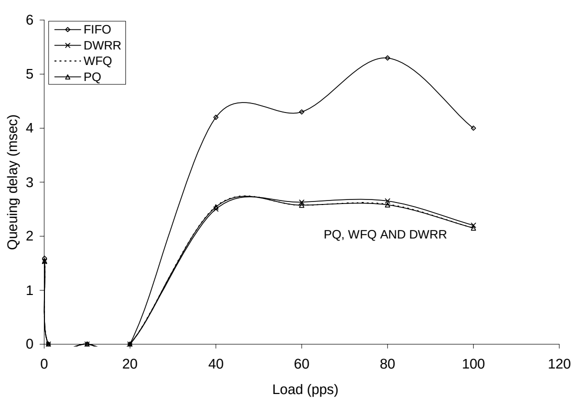

Fig. 6. Queueing delay versus load with RED policy.

4.2. Packet drop rate

As there is no differentiation between different classes, FIFO has the highest packet drop rate as demonstrated in Fig. 7. PQ, WFQ and DWRR give better (although on finer scale, DWRR is once again better as shown in Fig. 8). RED gives better results relative to drop-tail policy in all four disciplines; see Fig 9.

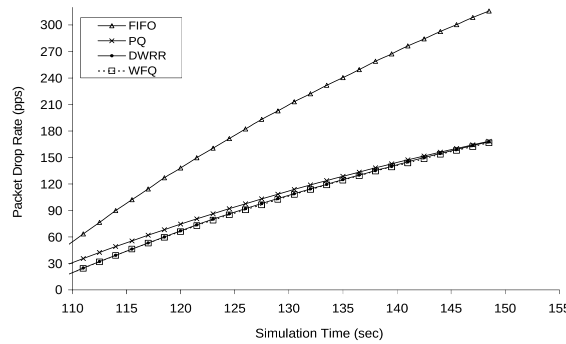

Fig. 7. Packet drop rate with drop-tail policy.

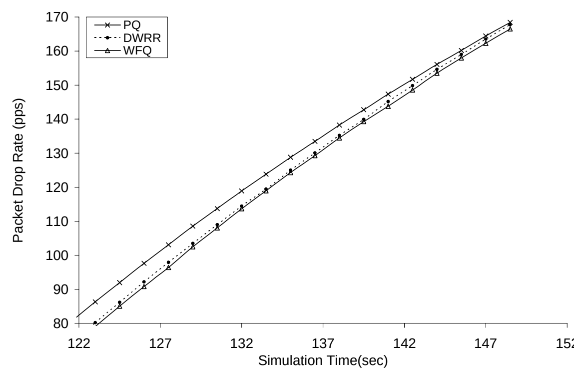

Fig. 8. Packet drop rate on fine scale.

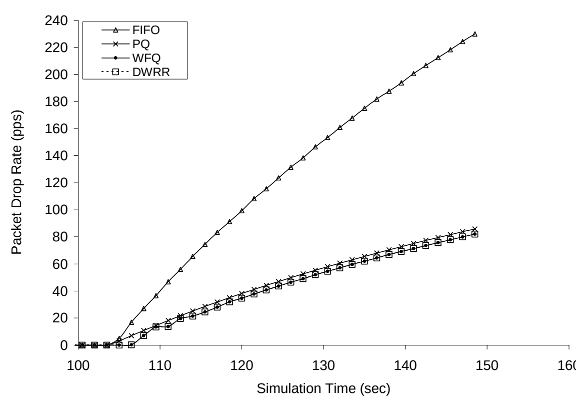

Fig. 9. Packet drop rate with RED policy.

Figure 10 and 11 show the packet drop rate versus load for drop-tail and RED policies respectively. At some points, packet drop rate is higher in case of RED which sounds logical as RED relies merely on probabilistic packet drop to avoid congestion and starvation problem.

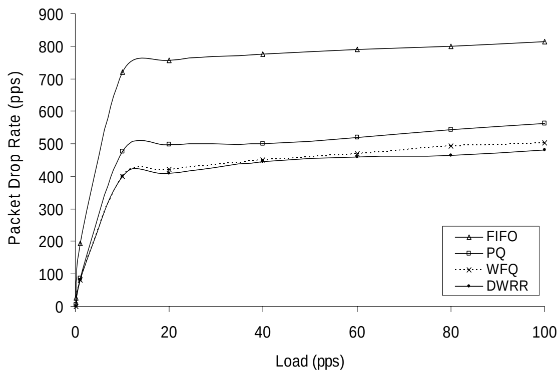

Fig. 10. Packet drop rate versus load with drop-tail policy.

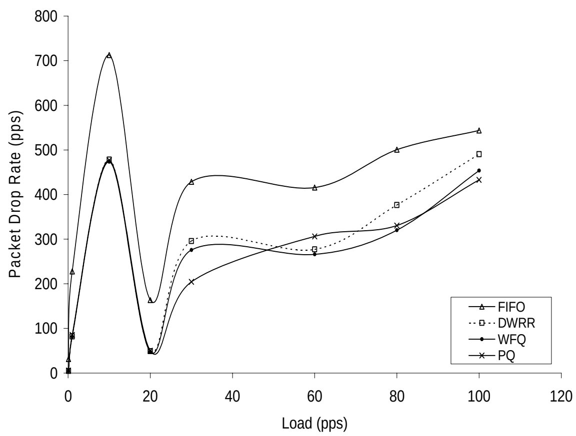

Fig. 11. Packet drop rate versus load with RED policy.

4.3. End-to-end delay & delay jitter

Figure 12 and 13 show the end-to-end delay for video traffic vs. simulation time with drop-tail and RED policies respectively. Similar results are obtained for delay jitter for video traffic in Fig. 14 and 15. It is clear that using RED has improved the performance of all queueing disciplines. We also noticed that WFQ and DWRR do not produce good results for this type of performance measures when compared with even FIFO. They try to avoid congestion and maintain fairness and in doing so, delay jitter and end-to-end become less guaranteed. Also in our simulation model, PQ outperforms all other queueing disciplines in terms of end-to-end delay and delay jitter for video traffic.

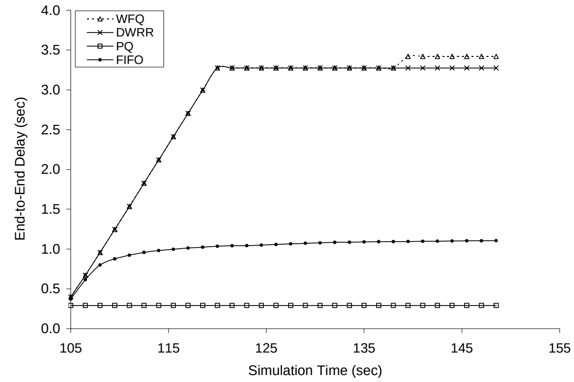

Fig. 12. End-to-end delay with drop-tail policy.

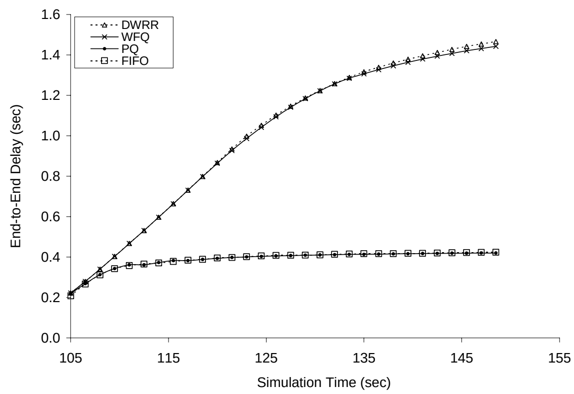

Fig. 13. End-to-end delay with RED policy.

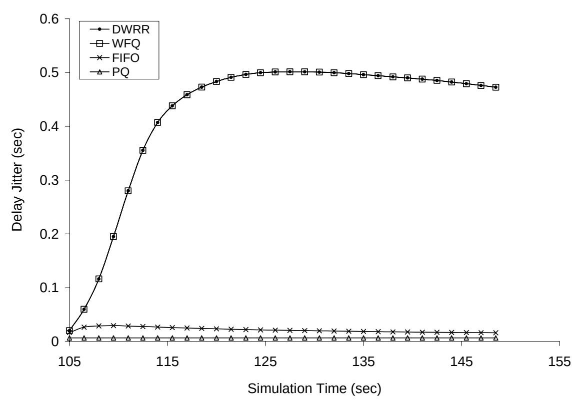

Fig. 14. Delay jitter with drop-tail policy.

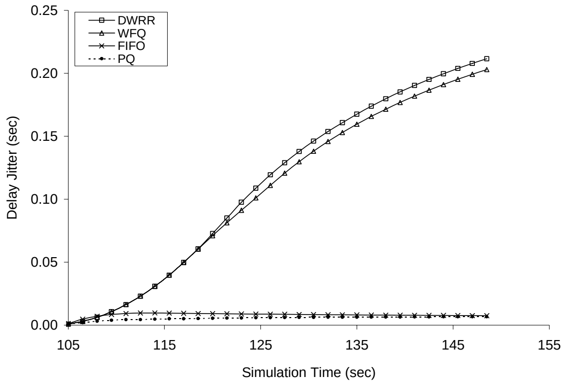

Fig. 15. Delay jitter with RED policy.

5. Conclusions

In this study we presented a simulation-based performance evaluation and comparison of four queueing scheduling disciplines for different traffic sources. We also examined the impact of using random-early drop as compared to drop-tail policy. The simulation results show DWRR that outperforms other disciplines in terms of average queueing delay and packet drop rate although PQ and WFQ are also very close to it for the considered traffic scenarios. But, it fails to produce acceptable results with respect to delay jitter and end-to-end delay for video traffic. In such case, PQ provides minimal end-to-end delay and delay jitter. We also noticed that using RED has greatly improved all the performance measures especially with FIFO. The reason is that RED monitors the average queue size and randomly drops packets when congestion is detected. Further work is still required to carefully examine these queueing disciplines and show the impact of self similar traffic and the use of traffic shapers at the edges of the network.

6. References

[1] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3/e. Addison Wesley, 2005.

[2] R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, June 1994.

[3] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, and E. Davies, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[4] C. Semeria, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines," White Paper, Juniper Networks, Inc., 2001.

[5] G. Bolch, S.Greiner, H.Meer, and K.S.Trivedi, *Queueing Networks and Markov Chains*. John Wiley and Sons, 1998.

[6] OPNET Technologies, Inc. http://www.opnet.com

[7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397-413, 1993.

[8] G. Mazzini, R. Rovatti, and G. Setti, "A Closed Form Solution of Bernoullian Two-Classes Priority Queue," *IEEE Communications Letters*, vol. 9, no. 3, pp. 264–266, 2005.

[9] A. Demers, S. Keshav, and S. Shenkar, "Analysis and Simulation of a Fair Queueing Algorithm," *Proceedings of the ACM SIGCOMM*, Austin, TX, pp. 1-12, 1989.

[10] J. Bennett and H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queueing," *IEEE INFOCOM´96*, San Fransisco, pp. 120-128, 1996.

[11] J. Bennett and H. Zhang, "Hierarchical Packet Fair Queueing Algorithms," *IEEE/ACM Trans. on Networking*, vol. 5, no. 5, pp.675-689, 1997.

[12] P. Goyal, H. Vin and H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," *IEEE/ACM Trans. on Networking*, vol. 5, no. 5, pp. 690-704, 1997.

[13] Y. Yen, W. Chen, J. Zhuang and H. Chao, "A Novel Sliding Weighted Fair Queueing Scheme for Multimedia Transmission," *Int. Conf. on Advanced Information Networking and Applications (AINA)*,vol. 1, pp. 25-30, 2005.

[14] T. Kwon, S. Lee and J. Rho, "Scheduling Algorithm for Real-Time Burst Traffic Using Dynamic Weighted Round Robin," *Proceedings of the IEEE Int. Symposium on Circuits and Systems*, vol. 6, no. 31, pp. 506 - 509, 1998.