Implementation of Traffic Conditioning and PHB mechanisms in OPNET

Evi Tsolakou, I. S. Venieris

National Technical University of Athens, Telecommunications Laboratory

Department of Electrical and Computer Engineering

Telecommunications Laboratory

9 Heroon Polytechniou str, 15773 Athens, Greece

tel.: + 30 1 772 2424 / fax: + 30 1 772 2534

email: evi@telecom.ntua.gr , ivenieri@cc.ece.ntua.gr

Abstract: *Differentiated Service Model (DiffServ) is currently a popular research topic as a low-cost method to bring QoS to today's Internet, especially in the backbone. The DiffServ architecture consists of two key components: traffic conditioning at the edge and Per Hop Behavior (PHB) at the edge and the core. Traffic Conditioners contain various elements such as classifier, meter, marker, shaper and dropper. The PHB mechanisms include the queue management and the scheduling.*

This paper addresses the impact of traffic conditioning and PHB mechanisms implemented in routers (edge) on the overall performance in Differentiated Services architecture, by using OPNET as the simulation tool. We have implemented the single and dual Token Bucket algorithms, the WRED/RIO queuing schemes with two sets of parameters (min, max, maxp), the RED dropping schemes, and the packet schedulers First In First Out (FIFO) and Weighted Fair Queuing (WFQ). A number of different Network Services has implemented, each one using a different traffic class. Each traffic class uses a different traffic profile and output queue. We investigate the effect of network transient, as expressed by changes in traffic load, on the performance of the scheduling algorithms and queuing/dropping schemes, in terms of both packet delay and queue size requirements. The simulation results are relative with the utilization on the backbone interface.

Keywords: Quality of Service, Differentiated Services, Traffic Conditioning, PHB Mechanisms, IP router, OPNET Simulation Tool

1 Introduction

The traditional Internet architecture offers best-effort service only. This service does not provide any means of preferentially treating traffic from customers who are willing to pay more. There is a distinct need to have a service that supports the wide range of applications on the Internet and the simplicity in forwarding mechanisms. Differentiated Services (DiffServ) [1] is an approach to IP QoS, which allows Internet Service Providers (ISPs) to support disparate applications and user expectations. The differentiated service model uses a combination of network edge elements (Traffic Conditioners & Per Hop Behaviors) and network core elements (Per Hop Behaviors) to achieve service differentiation. The edge and core elements are logically specified that give network operators the freedom to construct a wide set of services.
In the DiffServ model, packets entering a router, are first classified based on their e.g. source address/port, destination address/port. In sequence they experience traffic conditioning that involves metering, shaping and marking. After that, they are forwarded to the output interface of the router, where they experience a predefined PHB, e.g. EF (Expedited Forwarding), AF (Assured Forwarding), and BE (Best Effort). That involves DS-classification [2] and forwarding to a corresponding queue. The way these queues are handled is specified by the scheduling mechanism used. The packet scheduler is responsible for the order in which the packets of the various queues are dequeued and transmitted in the network.
In our model we use a set of flows with different type of service, which means that each flow corresponds to a different *traffic class* (TCL) [3] and is being transmitted to the different Token Buckets and Queues. Traffic classes can be viewed as the Network's mechanisms to implement the network services (NS) that are offered by the network provider to the customer. Five such network services have been identified: PCBR (Premium Constant Bit Rate), PVBR (Premium Variable Bit Rate), PMM (Premium MultiMedia), PMC (Premium Mission Critical) and Standard. The classes are called TCL1, TCL2, TCL3, TCL4 and TCL-STD respectively. Different applications are used for each traffic class, such as PCBR is used for voice, PVBR for video and interactive multimedia, PMM for low-quality video or file transfer, PMC for e.g. database queries and Standard for Best Effort traffic.
The rest of this paper is structured as follows: In section 2, an overview of traffic conditioning is given. The packet schedulers (FIFO, WFQ, PQWFQ) and queue management (RED / WRED-2 / RIO) algorithms are given in Section 3. Section 4 shows the structure of routers and the characteristics for each traffic class that we have implemented. In Section 5 the simulations results of the performance comparison are presented and finally Section 6 summarizes the major findings.

2 Traffic Conditioning

A traffic conditioner is referred to as a set of components that may include a meter, marker, shaper and a dropper. Traffic conditioners are usually located within DS boundary nodes (ingress, egress), but may also be located in nodes within the interior of a DS domain. The traffic conditioner bases its actions on the *traffic profile* that has been contracted between the user and the provider (as part of a static traffic conditioning agreement (TCA) or a dynamic reservation request). We illustrate the Traffic Conditioners with the Token Bucket profiles.

2.1 Meter/Marker/Shaper/Dropper

A meter meters each incoming packet and computes the resource consumption of the flow (aggregate) that the packet belongs to. The result is passed to the marker, which compares those measured properties to the traffic profile that has been contracted for the corresponding flow (aggregate). Depending on how the current measurements relate to the traffic profile packets are marked, i.e., the DS code point is (re) set. In OPNET Simulation Tool we (re)-set the Type of Service (ToS) Byte. Packets that get a token are said to be marked as *in-* profile and those that do not get token are said to be marked as *out-of-* profile [4]. In addition a dropper is implemented in order to drop packets (out-of-profile packets) and the shaper works within the Marker. The Token Buckets protects the small-window flows from packet loss by marking only *in-* profile for them [5]. The Traffic Conditioning functionality is depicted in Figure 2-1.

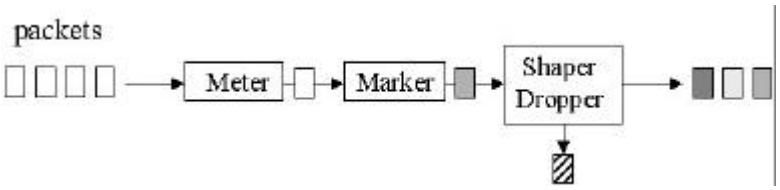

Figure 2-1: The Traffic Conditioning functionality.

For our purposes we use a single and a dual Token Bucket. The Token Bucket profiles are illustrated in OPNET Tool with the Committed Access Rate (CAR) algorithm.

3 PHB Mechanisms

After the packet has gone through the input interface of a router (CAR) and has not been dropped, it is inserted into the output interface of a router. This queue can be a simple queue holding all the traffic classes or a number of queues where each one holds one distinct traffic class. There are various queuing algorithms but only the ones implemented are described here. The PHB mechanisms include packet scheduling mechanism and queuing management.

3.1 Packet Scheduling Mechanisms

The implemented Packet Scheduling Mechanisms are the following:
First In First Out (FIFO) [6]: In FIFO, the packets that want to use an output link are placed into the output queue in the order in which they arrive. FIFO offers high cost-efficiency and no versatility.
Priority Queuing (PQ) [6]: The Priority Queuing algorithm dequeues the packet with highest priority. Then the packet with the next higher priority and finally the packet with the lowest priority. This algorithm has the following drawback: when packets from the highest priority are present, the packets in the other queues are probably starving. This is not compliant to the requirements of DiffServ. It is possible that when a source sends too much traffic and has the highest priority, may occupy up to 100% of the bandwidth. Therefore, the bandwidth should be managed by the SLA (Service Level Agreement) in order to prevent this behaviour.
Weighted Fair Queuing (WFQ) [6]: The Weighted Fair Queuing algorithm compares the weight for each subqueue with the bandwidth share. A packet from the queue with the biggest weight is dequeued. When this queue has no packet stored in it (no traffic present), the queue with the next bigger weight is dequeued. With this algorithm it is assured that no service occupies more bandwidth on the average than it should except. When there is enough bandwidth or when other services are present the excessive bandwidth is shared fair among the services. With WFQ a sharing is fulfilled.
Priority Queuing with Weighted Fair Queuing (PQWFQ) [3]: This algorithm is a combination of the described PQ and WFQ, as shown in Figure 3-1.

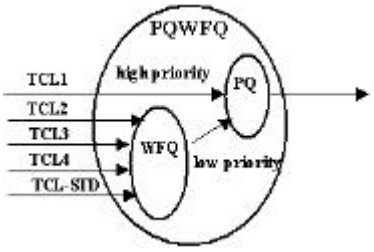

Figure 3-1: The PQWFQ algorithm.

The TCL1 traffic is dequeued with highest priority, no matter what other traffic is present, in order to support the stringent real-time constraints. The rest three traffic classes and BE traffic are dequeued with the WFQ algorithm. The TCL1 has a bandwidth share to which it is tested against, and though TCL1 has the highest priority

it cannot grab more than the bandwidth assigned to it. This mechanism allows the other classes to transit their traffic regularly.

3.2 Queuing Management

The traffic of each user is tagged as being *in* or *out* of their service profiles. Packets tagged as *in* profile are assigned lower drop precedence than those tagged as *out*-of-profile. In the OPNET Tool, the difference of the *in*-profile packets and the *out*-of-profile packets is determined by the Type of Service (ToS) Byte. We have implemented in the OPNET Tool, the RED, Weighted RED with 2 set of parameters (minth, maxth, maxp) and RED In & Out Profile Packets. These algorithms have been controlled and tested.

Random Early Detection (RED) [7][8][9]: RED allows a router to drop packets before the queue becomes saturated. RED achieves this by dropping packets with a certain probability depending on the average queue length (avg) as depicted in Figure 3-2.

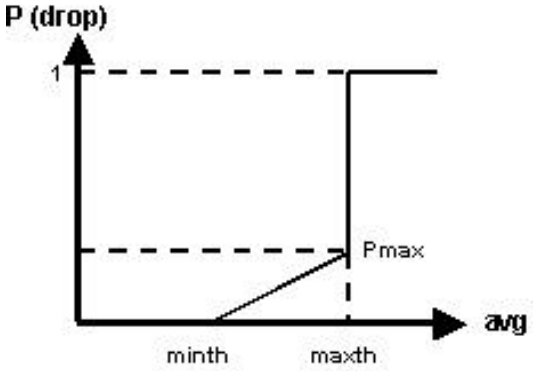

Figure 3-2: The RED mechanism.

The Pmax is determined from the Mark Probability Denominator Attribute at the OPNET Tool. The TCL-STD (Best Effort) uses the RED algorithm.

Weighted RED with 2 set of parameters (minth, maxth, maxp) (WRED) [10]: The WRED algorithm that we want to use is different than the one implemented in the OPNET Tool. The OPNET Tool has different minimum average queue size for each packet with different ToS Byte, but the same maximum average queue size and dropping probability (Pmax). This case is presented in Figure 3-3.

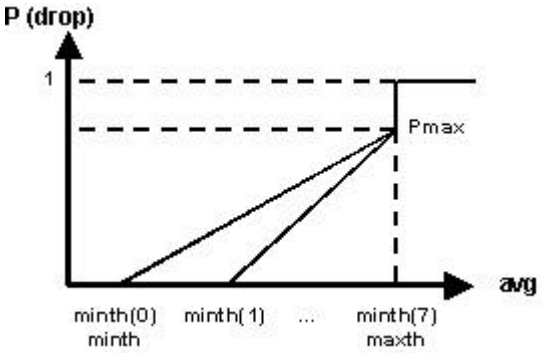

Figure 3-3: The WRED mechanism in OPNET.

The minth(0), minth(1) … minth(7) are the minimum average queue size for the packet with ToS Byte 0, 1, …, 7 respectively. The function that computes the minth, is given below:

$$\min\ th(ToS) = \min\ th + (\max\ th - \min\ th) * ToS / 7$$

Two separate levels of drop precedence can be supported with WRED-2 (Figure 3-4). We distinguish the *in*-profile packets from the *out*-of-profile packets with the ToS Byte. For that reason we set a new parameter that is

the ToS Byte. The *in*-profile packets represent the packets with this ToS Byte and the *out*-of-profile packets represent all the other packets.

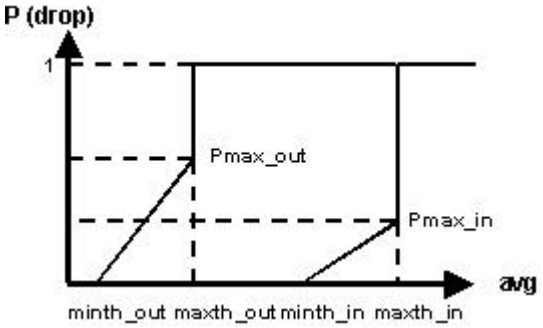

Figure 3-4: The WRED-2 mechanism.

In the OPNET Tool, we set the minth_in, maxth_in, Pmax_in and the ToS Byte for the *in*-profile packets and the minth_out, maxth_out, Pmax_out for the *out- of*--profile packets.
The TCL3 & TCL4 use this WRED-2 algorithm.

RED In & Out Profile Packets (RIO) [11]: The main difference between RIO and WRED-2 is that WRED-2 uses one average queue length to calculate drop probabilities, while RIO uses two average queue lengths. WRED-2 calculates its average queue length (avg) based on all packets present in the queue. RIO does that too but, in addition, it calculates a separate average queue length for packets in the queue tagged as *in*-profile (avg_in), see Figure 3-5.

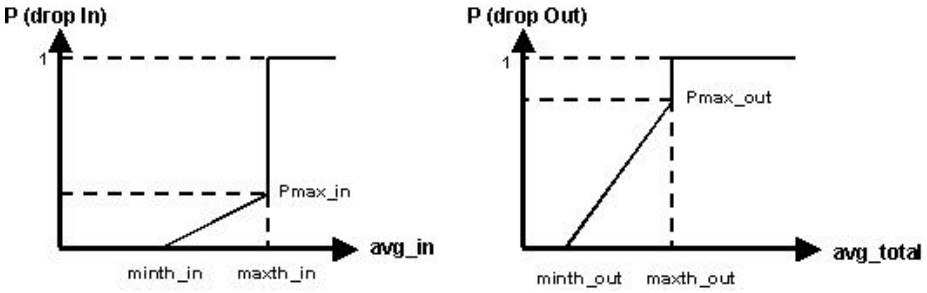

Figure 3-5: The RIO mechanism.

4 Implementation of Router's structure and Traffic Classes

We have used for the simulation topology Cisco routers and traffic generators corresponding to applications profiles. The QoS mechanisms in the routers involve schedulers (FIFO, WFQ, PQWFQ), buffer management algorithms (RED, WRED, RIO), Token Bucket Profiles (CAR) as explained in Figure 4-1. For each traffic class we have configured the edge routers (traffic conditioning, buffering, scheduling) and the core routers (buffering, scheduling) [3].

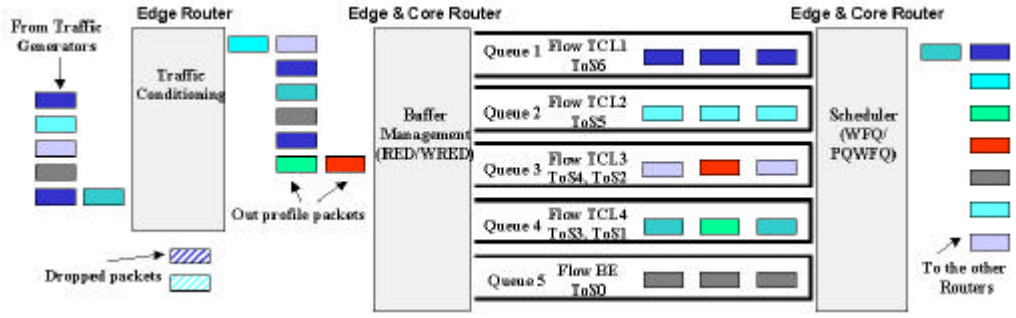

Figure 4-1: The Router's Structure.

Each TCL has the following characteristics:

- For TCL1 we use a Single Token Bucket (TB), which drops the out-of- profile packets. Packets of TCL1 are enqueued in the queue 1. The RED algorithm is employed as the queue management algorithm. We set for this TCL the ToS Byte to 6.
- For TCL2 we use the Dual TB, which drops the out-of-profile packets. Packets of TCL2 are enqueued in the queue 2. The RED algorithm is employed as the queue management algorithm. The ToS Byte is set to 5.
- For TCL3 we use a Single TB. Out profile packets are transferred with other Type of Service. Packets of TCL3 are enqueued in the queue 3. The WRED algorithm is employed as the queue management algorithm. We set for the *in*-profile packets the ToS Byte to 4 and for the *out*-of-profile packets to 2.
- For TCL4 we use a Dual TB. Out profile packets are transferred with other Type of Service. Packets of TCL4 are enqueued in the queue 4. The WRED algorithm is employed as the queue management algorithm. The ToS Byte for the *in*-profile packets is set to 3 and for the *out*-of-profile packets to 1.
- For TCL STD we do not use Token Bucket and the packets are enqueued in the queue 5. The RED algorithm is employed as the queue management algorithm. The ToS Byte is set to 0.

5 Simulations - Results

We have used a small-scale simulation topology with one Traffic generator, Background Traffic (BE) and two Edge Routers [12]. The link between the two Routers is of low capacity (for TCL1 Scenario is 150kbps or 200kbps or 250kbps or 300kbps, for TCL2 Scenario is 2.2Mbps, for All-TCLs Scenario is 1Mbps) and consists the bottleneck of the network.

5.1 TCL1 Scenario

In this scenario we use TCL1 that represents PCBR network service (voice applications) and background (Best Effort) traffic (Figure 5-1). A first case uses only a single FIFO and the second one uses for the TCL1 traffic conditioning (CAR) and a WFQ mechanism using the RED algorithm.
The flows for TCL1 that we use for the simulations have constant bit rate 54.4kbps, 88kbps, 120kbps, 200kbps and 256kbps with packet size 67.5B, 44B, 60B, 100B, 128B and 291B respectively.

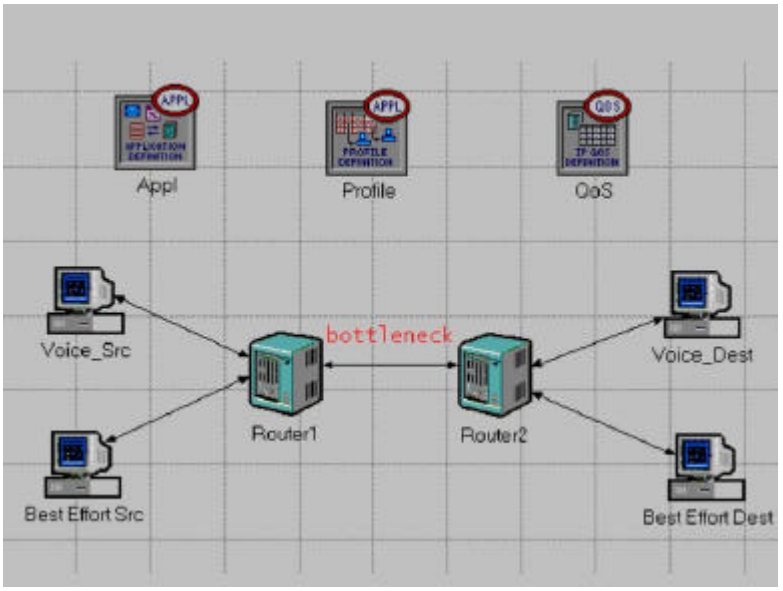

Figure 5-1: The Scenario for TCL1.

The maximum Queue Size for FIFO is set to 500 pkts. As depicted in Figure 5-2, the FIFO queuing delay of Router 1 for each flow used is bigger than 1 sec [13].

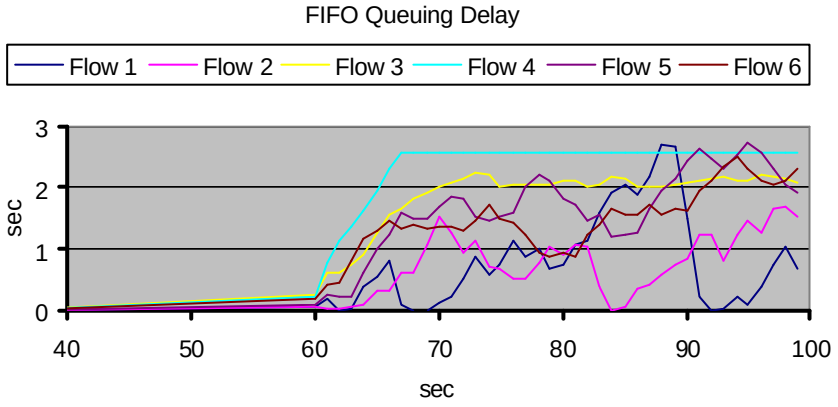

Figure 5-2: FIFO Queuing Delay for TCL1 Scenario.

In the second simulation we use a single Token Bucket with Peak Rate (PR) equal to 200kbps and Bucket Size for PR (BSP) equal to 256B. The weight of WFQ for the TCL1 is set to 60% and for the BE to 40%. The maximum queue size for TCL1 is set to 100 pkts and for BE to 500 pkts. The RED is enabled for each flow. The WFQ queuing delay for all flows of TCL1 from Figure 5-3 is less than 9 msec.

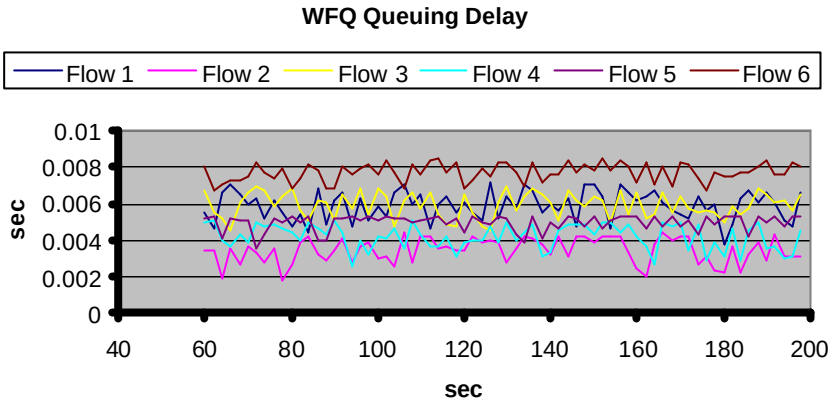

Figure 5-3: WFQ Queuing Delay for TCL1.

5.2 TCL2 Scenario

In the second scenario we use TCL2 that represents PVBR network service (video applications) and a Best Effort traffic (Figure 5-4). The first simulation has only a single FIFO and the second a traffic conditioning for the TCL2 is used and a WFQ mechanism with the RED algorithm for each flow.
The flows for TCL2 used for the simulations have constant bit rate 1.64Mbps and 2Mbps with packet size 2050B and 5150B respectively.

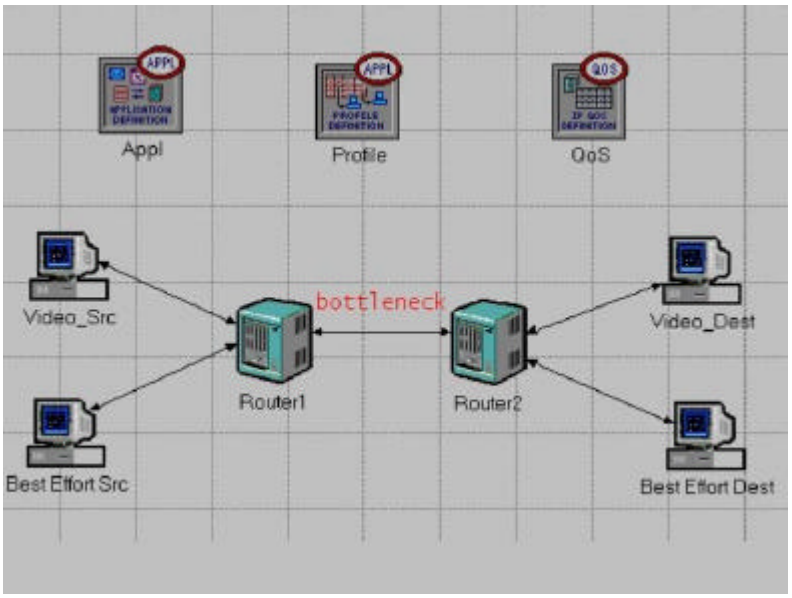

Figure 5-4: TCL2 Scenario.

The maximum Queue Size for FIFO is set to 500 pkts. The FIFO queuing delay for both cases of TCL2 is bigger than 3 sec (see Figure 5-5).

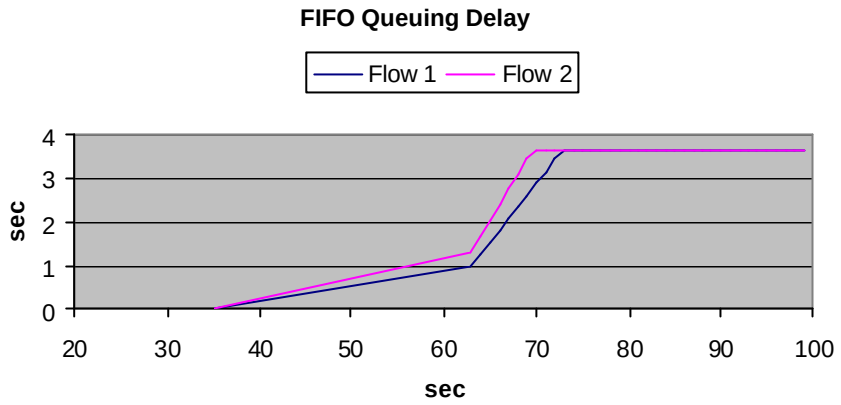

Figure 5-5: FIFO Queuing Delay for TCL2 Scenario.

In the second simulation we use a dual Token Bucket, where the Sustainable Rate (SR) is set to 1Mbps, the Bucket Size for SR (BSS) to 5120B, the Peak Rate (PR) to 5Mbps and the Bucket Size for PR (BSP) equal to 1024B. The weight of WFQ for the TCL2 is set to 60% and for the BE to 40%. The maximum queue size for TCL2 is 100 pkts and for BE is 500 pkts. The RED is enabled for each flow. The WFQ queuing delay for all flows of TCL2 is less than 40 msec (see Figure 5-6).

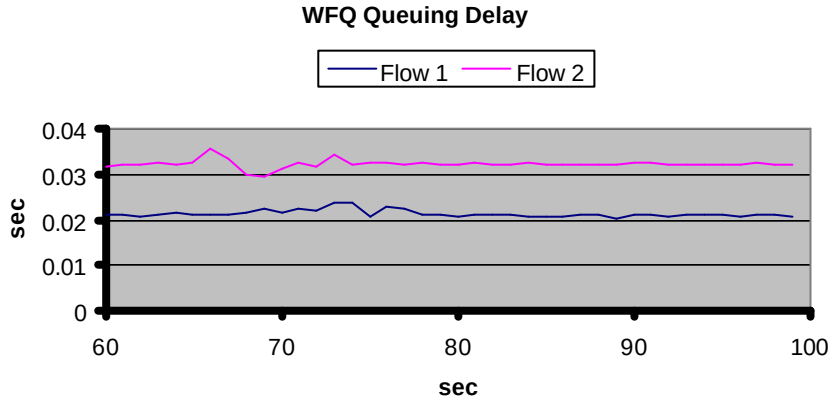

Figure 5-6: WFQ Queuing Delay for TCL2.

5.3 Scenario with all TCLs

In the first case, all flows are enqueued in a single FIFO queue and no queue management algorithm is taking place. In the second a Traffic Conditioner is taking place for each TCL and then the flows are enqueued in a WFQ queue with 5 sub-queues and RED or WRED is employed as the queue management algorithm.

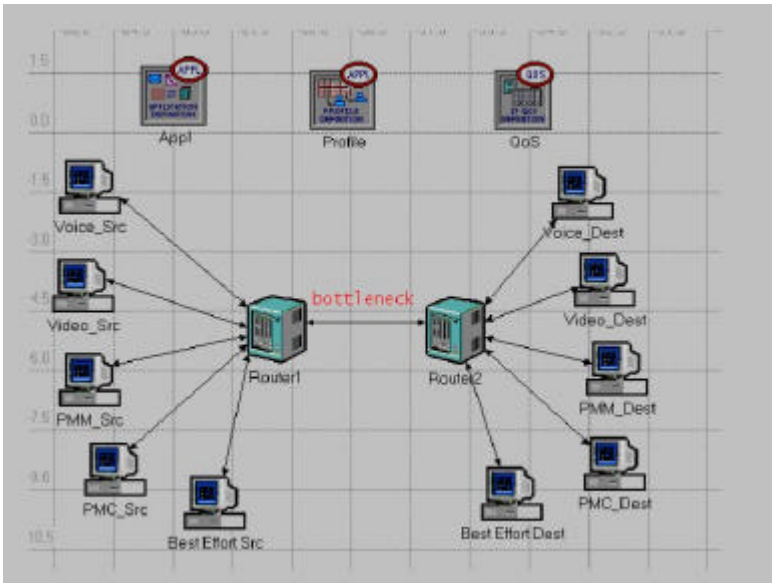

Figure 5-7: Scenario with all TCLs.

The characteristics for each TCL are described below:
TCL1: Constant Bit Rate 32kbps, Packet Size 128B, Single Token Bucket with PR=32kbps and BSP=256B.
TCL2: Constant Bit Rate 600kbps, Packet Size 1024B, Dual Token Bucket with SR=1Mbps, PR=5Mbps, BSS=2048B and BSP=1024B.
TCL3: Constant Bit Rate 60kbps, Packet Size 3125B, Single Token Bucket with SR=100kbps and BSS=15000B.
TCL4: Constant Bit Rate 200kbps, Packet Size 2048B, Dual Token Bucket with SR=5kbps, PR=50kbps, BSS=10240B and BSP=2048B.
TCL-STD: Constant Bit Rate 822.4kbps, Packet Size 1024B, no Token Bucket.
The maximum Queue Size for FIFO is set to 1000 pkts. The FIFO queuing delay for all TCLs is about 8 sec (see Figure 5-8).

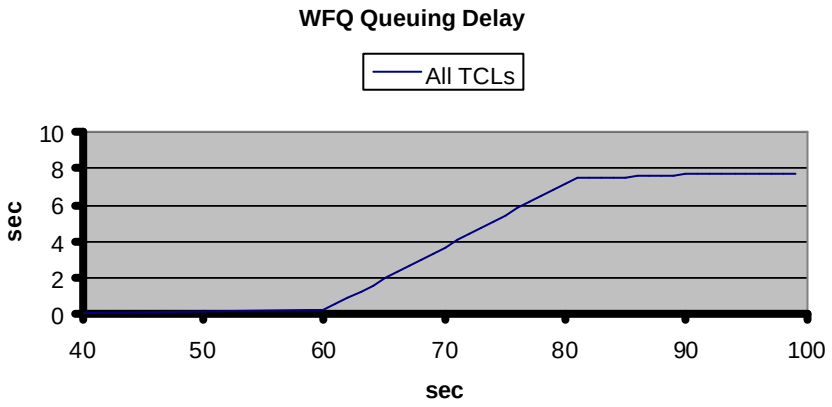

Figure 5-8: FIFO Queuing Delay for All TCLs Scenario.

In the second simulation the weight of WFQ for the TCL1 is set to 50%, for TCL2 to 20%, for PMM to 15%, for PMC to 10% and for BE to 5%. The maximum queue size for TCL1 and TCL2 is 100pkts, for TCL3, TCL4 and BE is 500 pkts. The RED is enabled for each flow. As we can see the WFQ queuing delay for TCL1 is less than

8msec, for TCL2 is less than 20msec, for TCL3 is less than 50msec and for TCL4 is less than 80 msec (see Figure 5-9).

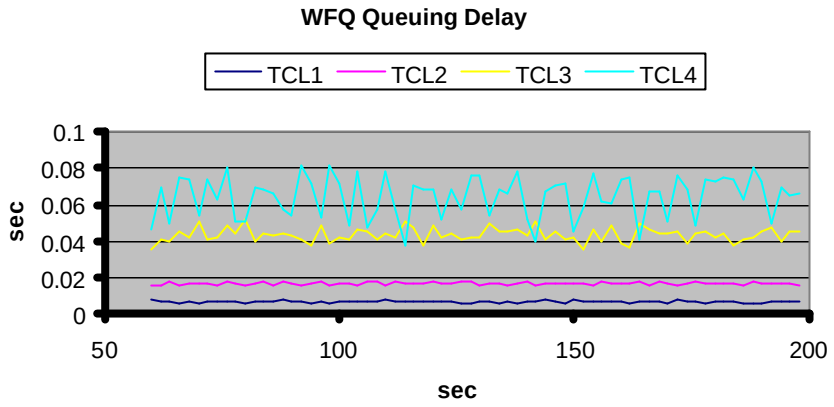

Figure 5-9: WFQ Queuing Delay for TCL1, TCL2, TCL3 and TCL4.

6 Conclusions

In this paper we have implemented and evaluated the Traffic Conditioning (Token Bucket Profiles), some Packet Schedulers (FIFO, WFQ, PQWFQ) and buffer management (RED/WRED/RIO) algorithms, in offering sheltering under different loads with background traffic (Best Effort). Through these simulations, we verified the correctness of our design and implementation. We show that the FIFO queuing delay in all scenarios is much bigger than the WFQ ones. The basic reason is that, in the FIFO all packets (from different classes) are enqueued to the same queue, therefore all TCLs has the same queuing delay with the BE ones. The WFQ algorithm gives a separate queue and different weight (priority) to each TCL and to the BE traffic. In this case, the queuing delay is different for each TCL (queue) and is independent of the other TCLs and background traffic. For that reason, the WFQ algorithm has much less queuing delay for the TCLs than the FIFO ones.

REFERENCES

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

[2] K. Nichols, S. Blake, F. Baker and D. L. Black, "Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers", RFC 2474, December 1998.

[3] Deliverable D1301, "Specification of traffic handling for the first trial", AQUILA project consortium, June 2000.

[4] A. Dobreff, "Comparison of Simulation and Real Functionality for the Mapping of Differentiated Services to ATM", Diploma of the Faculty of Philosophy and Science of Nature University Bern, Switzerland, 1999.

[5] A. Feroz, A. Rao and S. Kalyanaraman, "A TCP-Friendly Traffic Marker for IP Differentiated Services", IwQoS'2000, Pittsburgh, June 2000.

[6] M. Markaki, M. P. Saltouros and I. S. Venieris, "Comparison of Packet Schedulers for Differentiated Services in the Internet", Proc. IFIP ATM & IP 2000, Ilkley, U.K, July 2000.

[7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol.1 n.4 pp. 397-413, August 1993.

[8] M. May, T. Boland and J.-C. Bolot, "Analytic Evaluation of RED Performance", in Proceedings of Infocom 2000.

[9] S. Cnodder, O. Elloumi and K. Pauwels, "RED behavior with different packet sizes", in Proceedings of the Fifth IEEE Symposium on Computers and Communications, vol. 3, p.p. 793- 805, July 2000.

[10] U. Bodin, O. Schelen and S. Pink, "Load-tolerant Differentiation with Active Queue Management", SIGCOMM Computer Communication, vol.30 n.3, July 2000.

[11] D. Clark, W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service", IEEE/ACM Transactions in Networking, vol. 6, n. 4, pp. 362-373, August 1998.

[12] J. Wang, K. Nahrstedt and Y. Zhou, "Design and Implementation of DiffServ Routers in OPNE", in Proceedings of OPNETWORK'00, Washington D.C., Aug 28 - Sep. 1, 2000.

[13] V. Firoiu and M. Borden, "A Study of Active Queue Management for Congestion Control", in Proceedings of IEEE/INFOCOM, 2000.