Packet Scheduling Algorithms Simulator WFQ, SCFQ And WF2Q

Muhilan R, Arulselvi S, Kiran Kumar T.V.U

Abstract: As the Internet evolves into the global communication infrastructure, there is a growing need to provide a broad range of QoS guarantee for different applications, which brings forth the necessity of traffic management. The scheduling algorithm in the router determines how packets from different connections interact with each other by controlling the order in which packets are served. As one of the most important parts in traffic management, the packet scheduling plays an important role in the QoS provisions. The Packet Scheduling is the shared transmission resources should be intentionally assigned to some of the users at given time. The process of assigning users packets to appropriate shared resources to achieve some performance guarantee is so called "Packet Scheduling", which is done by the fair queueing methods.

Index Terms: Packet Scheduling, Scheduling Algorithms, Fair Queuing, Generalized Processor Sharing, WFQ, SCFQ, WF2Q, Congestion Control.

1 Introduction

FAIR Queueing and its variants serve as a congestion control scheme in datagram networks. Fair Queueing provides several important advantages over the usual FIFO queueing algorithms such as fair share of bandwidth allocation, lower delays for sources that use less than their fair share of bandwidth and protection from ill-behaved sources. Due to certain problems in Fair Queueing enhanced versions of Fair Queueing algorithms were developed such as Weighted Fair Queueing and Worst Case Fair Weighted Fair queueing that emulates the ideal Fluid Generalized Processor Scheduling algorithm. These algorithms provide fairness even under worst conditions while providing QoS and absolute fairness guarantee. The objective of the research is to develop an analysis and comparison tool for various Fair Queueing schemes that will allow end user to analyze various parameters such as packet latency, delay characterstics, packet interleaving at router output, etc. for real time configurable source(s). Having studied the basic principle of internet working and the protocols used in the internet and about the packet scheduling algorithm used for the packet scheduling. We have conducted a simulation study of selected scheduling disciplines. In Packet scheduling algorithms some are shortlisted for this project with the basic parameters. For this we have completely studied about the packets and Tcp/Ip flow, Buffer management and many more. The Simulation for this project has done in Python language. So the language Python also studied throughout the development of the phase. First, here the basic parameters of the algorithms had been short out. Using these parameters, hierarchy algorithm of the simulator program has developed. The simulator is designed minimally.

- *Department of Electronics and communication Engineering,Aksheyaa college of Engineering, Puludivakkam, Maduranthagam Kancheepuram– 603314, Tamilnadu, India.*
  Email : mrmuhilmit@gmail.com
- *Department of Electronics and communication Engineering, Bharath University, Chennai – 600073, Tamilnadu, India*

2 Scheduling Algorithms And Queueing Theory

2.1 The Need of Fairness

Scheduling is needed whenever there is contention for a shared output channel. Desired characteristics of scheduling algorithms are that they should provide lowest possible fairness bound with respect to ideal fluid system and can be easily implemented in hardware. The interleaved schedule should also guarantee that packets are processed within bounded interval based on either the requested rate or on the average rate allotted per session. Elaboration of comparison characteristics in given in the following section.

2.2 Comparison Criteria

1. Fairness: This relates to the bandwidth occupancy by a particular source in relation to the requested or shared bandwidth of router. High Fairness is preferable.
2. Fairness Bound: The time required by scheduling algorithm to match defined fairness standard. Lower value for Fairness Bound is preferable.
3. Delay Bound: Packet in Router queue must be processed within specifiable delay.
4. There should be no bias towards large or small packet size.

2.3 Generalized Processor Sharing

Generalized Processor Sharing is a work conserving no.1 discipline. This is the ideal form of packet scheduling algorithm but it cannot be implemented in hardware since it assumes that packets flow as fluid stream i.e. infinitely small portion of packet can be processed simultaneously and multiple sources can be processed at the same time Packet Processing Time: Each packet is processed in L_i^k/r_i where is L_i^k the length (size) of packet k of flow i and is the assigned rate r_i to flow i.

2.4 Weighted Fair Queueing

WFQ is based on GPS discipline and it tries to emulate GPS on a Packet by packet basis (it is also called PGPS Packet Generalized Processor Sharing). It can be used to provide QoS based services. Implementation of WFQ requires Virtual Clock and each packet is assigned a Virtual Finish Time. It emulated GPS by selecting next packet to service by selecting the packet with Smallest Virtual Finish Time First.

Virtual Finish time of packet *k* of flow *í* is given as

$Fí^k = Sí^k + Lí^k/rí$

where $Sí^k$ is the Virtual Start Time of packet *k* of flow *í*.

Virtual Start Time of packet *k* of flow *í* is given as

$Sí^k = max\ (Fí^k, V(aí^k))$

where $V(aí^k)$ is the Virtual Time of Server at arrival of packet a which is k th packet of flow i.

2.5 Worst Case Weighted Fair Queueing

WF2Q tries to fix the anomalies in WFQ by using SEFF Smallest Eligible Virtual Finish Time First. It accurately simulates GPS system by selecting packet both on basis of Virtual Start and Virtual Finish Time. WF2Q finds next packet to process only from list of packets that have been started to process in associated GPS system. An important point to note is that even though WF2Q does not serve packets that have Virtual Start Time greater than current Virtual Clock time; it is still a work conserving scheduler as it processes packet when corresponding GPS system is processing packet therefore matching GPS schedule. Since GPS is work conserving, hence WF2Q is also working conserving. WF2Q satisfies all the requirements for fairness, protection and non-biasness however as with WFQ, it is com-puting intensive and requires memory to maintain state of server.

Virtual Finish time of packet *k* of flow *í* is given as

$Fí^k = Sí^k + Lí^k/rí$

where $Sí^k$ is the Virtual Start Time of packet *k* of flow *í*.

Virtual Start Time of packet *k* of flow *í* is given as

$Sí^k = max\ (Fí^k, V(aí^k))$

where $V(aí^k)$ is the Virtual Time of Server at arrival of packet a which is k th packet of flow i.

2.6 Self Clock Fair Queueing

Each arriving packet is tagged with a service tag before placed in to queue. The packets are served in increasing order of their service tag.

$Fí^k = Lí^k/rí + max\ (F(i\text{-}1)^k, V(ai^k))$

v(t) is defined as the service tag of the packet receiving service at the time t.

$Ví^k = Fí^k$

When the busy period is over v(t) and session packet counters are initialized to 0.

SCFQ system is also work-conserving.

3 Simulator Design

3.1 Basic Working

The simulator that we designed performs the following tasks. It takes 3 command line arguments. First is a number from 1 to 4 telling which scheduling algorithms to simulate

1 for WFQ
2 for WF2Q
3 for SCFQ
4 for simulating all

Second argument is the name of file containing information about arriving packets Third argument is the name of file containing information about queues like their weights

3.2 Input Files

Pkt.conf

This file contains packet information, first column being the packet arrival time, second is the queue name on which it arrived and third is the packet length.

Queue.conf

This file contains information about queues, first colum being the queue name and second is its weight.

3.3 Output Files

On execution simulator will generate data files and graphs depending upon the command line argument. Output files for different scheduling algorithm will contain comma seperated following information:

- Packet and queue no.
- Arrival time
- Length
- Start and finish time in a particular scheduling algorithm

3.4 Assumptions and Limitation

One has to provide a time resolution and accuracy of the simulation depends on this time resolution. In simpler words our real time can increase by 1s or 1ms or 1ps and so on , higher the resolution better the accuracy. Output link bandwidth is assumed to be 1 unit per second. All floating point comparisons are done with 6 digits after decimal accuracy.It may be possible that a packets finish number may not match exactly the simulated round number, in that case the packet is assumed to finish once the simulated round number is greater than the packet finish number.

3.5 Implementation And Architecture

3.5.1 Classes, Their Attributes and Methods

We have two classes namely packet and queue.

Packet:

Its attributes are arrival time, length, finish number, queue name and start and finish time for 4 of the scheduling algorithms. This class also provides methods for printing and converting a packet into string , which can be used fot logging a packet.

Queue:

Its attributes are queue name, is backlogged , packet list, weight, max finish number and next finish number. This class provides methods for dequeue and enqueue packets. These methods will be used to simulate GPS scheduling. These two methods will also update the queue status such as is backlogged or not and next finish number in the queue. Each queue will have have list of packets that are arrived so far.

3.5.2 Model

First we use a lop which simulates the real time by increasing the counter rt by given resolution each time. Inside that loop i.e at every instant of time we do following activities: check for new pkt arrival, If found calculate its finish no. and add it to the proper queue. Check whether any pkt has finished by matching the pkt finish no. with simulated round no., if any pkt is finished we remove that from the queue. Now we find out the no. of active queues. If the WFQ has finished a pkt or is idle pick a new pkt from the backlogged queue according to WFQ algo. Update the simulated round no. according to the no. of active queues.

For SCFQ

Check for arrival of a new pkt i.e if our counter rt matches arrival time of any of the packet in the file. Then pick a new pkt having the lowest finish no. Set the current round no. to newly picked pkt's finish no.

4 Results And Analysis

From our simulation we now have arrival time, the time when a packet start getting service, finish time when processed using GPS, WFQ, WF2Q and SCFQ. We calculated delay and latency from all of this data and then plotted some graphs. This example is taken such that the purpose of WF2Q can be visually explained. This example has eleven queue starting from A to K. A has ten times higher weight than other queues and all other have same weight. At time zero, ten packets arrive at queue A and one packet in each of other queues.

4.1 WFQ

Incase of WFQ all packets from A are scheduled before any packets from other queues. This happens because of higher queue weight and equal length packets. Packets from other queues will starve till packets from A finishes. Fig1shows this scenario where packets from A are scheduled first.

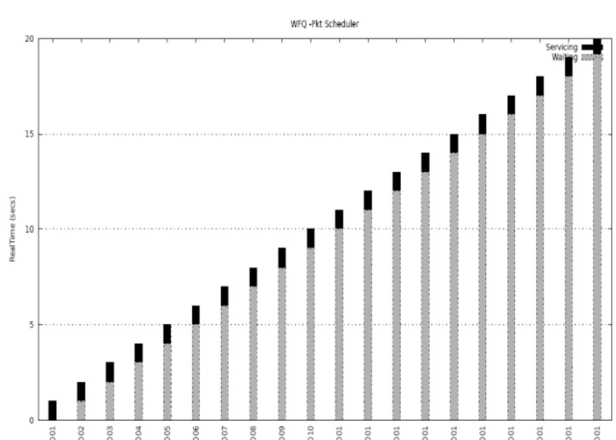

Fig. 1. WFQ – Pkt Scheduler

4.2 SCFQ

For this example SCFQ produces a same schedule as WFQ, but the simulator can also be used to find out the maximum service lag that could occur between WFQ and SCFQ by choosing the proper packet arrival pattern

4.3 WF2Q

In case of WF2Q not all packets from A are scheduled before other packets , because WF2Q checks whether a packets has started servicing in GPS before picking a packet, so other queues doesn't starve like they do in WFQ. Fi2 2depicts this case where scheduling happens in A,B,A,C ... pattern.

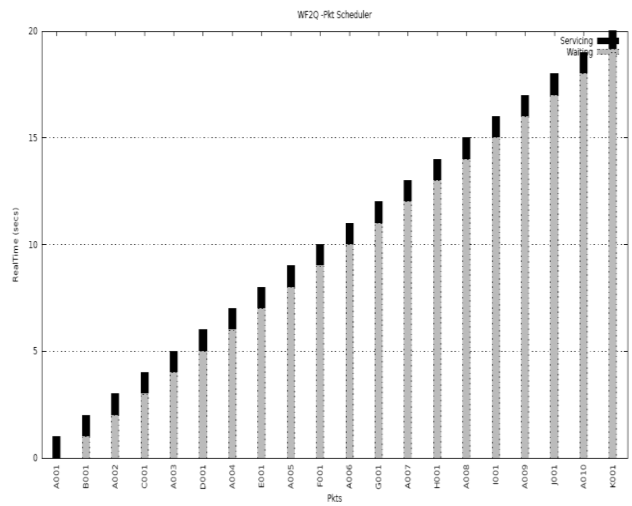

Fig. 2. WF2Q Pkt Scheduler

5 Comparisions Of Delays

Fig 3. also depicts the behaviour of WFQ and WF2Q. In WFQ packets from A suffers a linear delay since it is coninuously backlogged. So that all other queues wait for packets in A to finish. But in case of WF2Q , packets from queue A and other queues share the output link.

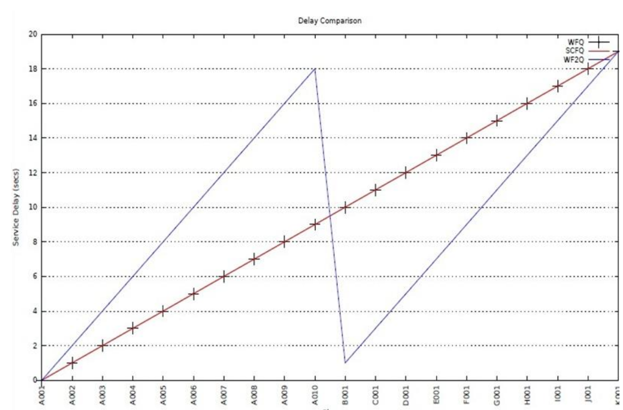

Fig. 3. WFQ,WF2Q,SCFQ Delay Comparison

6 Conclusion

The scheduler is useful in analyzing the scheduler for these worst case behaviors by choosing proper test scenarios. This also gives insight about how exactly round numbers are assigned (especially in SCFQ) and shows the ease of SCFQ implementation when compared to WFQ and WF2Q. Though this scheduler was not intended to handle huge number of packets, still it can function with huge number of packets with considerable performance. This implementation can be optimized at places like rather than updating round number in each run, it can be updated only on the event of packet arrival and packet departure. This can also be used to check various behaviors such as fairness, isolation for scheduling algorithms. This is feasible because all statistics about the packets are captured in the results file.

REFERENCES

[1] S.J. Golestani, A self-clocked fair queueing scheme for broadband applications INFOCOM '94. Networking for Global Communications., 13th Proceedings IEEE

[2] A. K. Parekh and . R. G. Gallager. A general- ized processor sharing approach to flow control in integrated services networks. In Proc. IEEE INFOCOM92,

[3] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queuing algorithm,” Journal of Internetworking Research and Experience, vol. 1, no. 1, pp. 3–26, Sept. 1990.

[4] J. R. Davin and A. T. Heybey, “A simulation study of fair queueing and policy enforcement,” SIGCOMM Comput. Commun. Rev., vol. 20, no. 5, pp. 23–29, 1990.

[5] jon C.R. Bennett and H. Zhang, "Hierarchical Packet Fair Queueing Algorithms". IEEE/ACM Transactions on Networking, 5(5):675-689, Oct 1997. Also in Proceedings of SIGCOMM'96, Aug, 1996

[6] Introduction to Packet Scheduling Algorithms for Communication Networks -- Tsung-Yu Tsai1, Yao-Liang Chung2 and Zsehong Tsai2