Advances in Computer Science and Engineering
Volume 7, Number 2, 2011, Pages 119-125
Published Online: December 2011
Available online at http://pphmj.com/journals/acse.htm
Published by Pushpa Publishing House, Allahabad, INDIA

A SURVEY OF PRIORITY QUEUEING (PQ) ALGORITHMS IN COMMUNICATION NETWORKS

S. Nandhini and S. Palaniammal

Department of Mathematics
Garden City College of Science and Management Studies
Bangalore-560049, India

Department of Science and Humanities
VLB Janakiammal College of Engineering and Technology
Coimbatore-641042, India

Abstract

Queueing theory plays an important role in computer networks. In computer networks, when data packets are sent out from a host, they enter a queue where they wait for processing by operating system. Queueing models and their concepts are used to process the data packets in the queues. In this paper, we are going to analyze about priority queueing model and priority queueing (PQ) algorithms.

1. Introduction

In network communication, optimizing and managing the queueing of data packets flow is an important subject for traffic flow management, resource allocation and QoS (quality of service). In a network, buffers are used for collection of data packets waiting to be processed. Many queueing models and queueing algorithms are used to process these packets efficiently. Most of the queues in networks are processed on FIFO (First In First Out) basis. But there are also other queueing models such as LIFO (Last In First

Keywords and phrases: priority queueing, QoS, CoS.

Received September 20, 2011

Out), random servicing (RS) and priority queueing (PQ), fair queueing (FQ), weighted fair queueing (WFQ). This paper deals with concepts of priority queueing, starvation and PQ algorithms.

2. Prioritization of Queues

In communication networks, QoS (quality of service) and CoS (class of service) play an important role. QoS refers to the capacity of a network to provide better service to selected network traffic over various technologies. QoS offers a better service by assigning a higher priority for a particular flow of traffic or limiting the priority of another flow. CoS is a traffic management mechanism in networks which groups the similar traffic together as one class and priority is assigned for that class. Prioritization of queues plays a vital role in QoS and CoS technologies.

In priority queues, data packets that contain important data (may be information about network status) or the packets that must arrive in a timely way would be given higher priority and would be transmitted ahead of lower priority data. Priority queueing model has further two refinements [1] namely preemption and non-preemption.

In preemptive cases, the higher priority customer will be allowed to enter the service immediately regardless of the lower priority data which is already present in the service. In non-preemptive cases, when the higher priority customer arrives, he goes ahead of the queue and will wait for lower priority which is in service, is completed without interrupting it.

Basically there are four queues of traffic in priority queueing viz. high, medium, normal and low which is based on the importance of data packets. Here it works as, as long as the high priority jobs are there, medium is neglected and medium priority jobs will be processed after all high priority packets are processed and similarly as long as medium priority jobs are there the other two will be neglected till the medium is completed. Figure 1 shows how priority queues work.

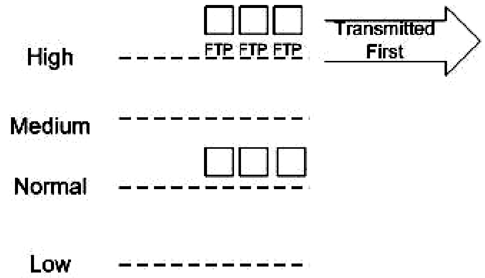

Figure 1

In priority queueing, if anytime a packet with higher priority enters the queue, then the other priority packets which are in the process, are stopped and the high priority packets are processed first as shown in Fig. 2.

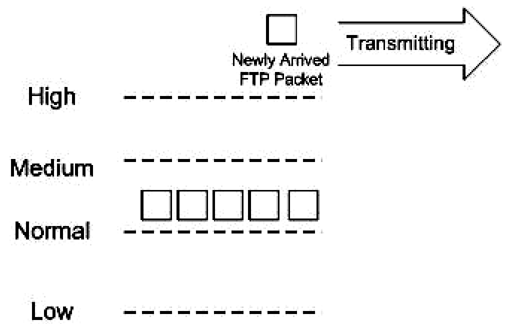

Figure 2

It sometimes happens that if all the data packets of higher priority are processed first, the packets of medium or normal or lower priorities may be dropped without processing. This leads to loss of data and this refers to starvation.

The algorithms we are going to deal gives efficient ways of priority queueing scheduling and also solve the problem of starvation.

3. Priority Self-detective Random Early Detection (PSRED) Algorithm

This algorithm deals with the realization [3] for self-adaptive management mechanism of dynamic packet priority and realization of self-adaptive mechanism of packet loss for ad-hoc networks. In this algorithm, as soon as the high priority packets have been served for time ‘*t*’, the inferior priority queue starts upgrading the priority of its packets from the beginning of the queue.

In networks the loss probability of the packets depends on the length of the queue. If the queue is longer, then the loss probability is more. In this algorithm, when the average queue length is updated, a self-adaptive adjustment algorithm starts which helps to achieve the smallest packet loss probability. This algorithm serves low priority packets appropriately when the highest priority packets are served which avoids “starving to death” for inferior queues and the packet loss probability is very less when this algorithm is applied.

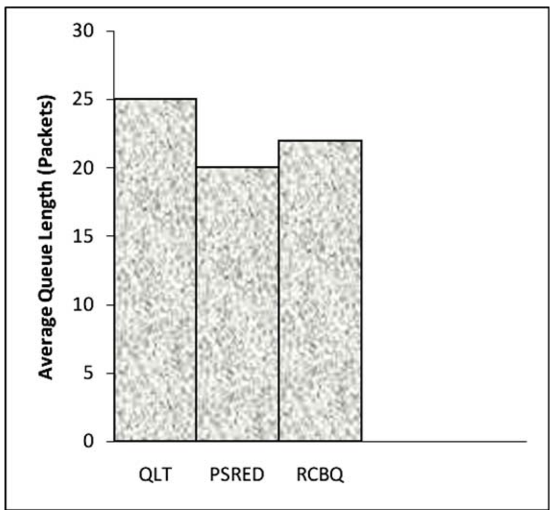

Figure 3

PSRED algorithm is implemented in NS-2 and a comparison is made with QLT and RCBQ algorithms. The result shows that (i) the probability of loss of packets is less due to self-adaptively of PSRED. It controls the packets which will control the loss probability of packets of high priority queue in small range (ii) the transmission delay of the data packets are reduced.

4. A Distributed Algorithm for Low Priority Data Transfer

TCP-LP (low priority) algorithm [2] achieves two class service prioritization without any support from the network. The objective of this algorithm is to (i) utilize the unused bandwidth and (ii) also being transparent to TCP. TCP-LP allows low priority applications to use all excess capacity while also remaining transparent to TCP. TCP-LP achieves two-class prioritization without any support from the network. One class of applications of TCP-LP is a low priority file transfer over an internet. A second class of TCP-LP deals with bandwidth monitoring, performance optimization and performing statistical inference on delay measurement. TCP-LP transmits the data at the rate of available bandwidth. According to the change in the bandwidth over time, TCP-LP has a mechanism to adapt to the changing network environment.

In this algorithm, a reference model is created with two levels as follows. First level is to maintain TCP-LP transparency to TCP and the second level is to maintain TCP fairness among multiple TCP-LP flows competing to share the excess bandwidth.

TCP-LP employs two important mechanisms. First, TCP-LP flows must detect the oncoming congestion prior to TCP flows. TCP-LP follows one-way delay measurements so that it detects congestion from source to destination and prevents false early congestion indications from reverse cross-traffic. The second is congestion avoidance policy of TCP-IP which comprises of (i) fast withdrawal in presence of congestion in TCP flows (ii) utilizing the excess bandwidth (iii) achieving fairness among TCP-LP flows.

When this algorithm is deployed in NS-2 simulator, it shows that TCP-LP is largely non-intrusive to TCP flows and it also shows that single and aggregate TCP-LP flows tracks and utilizes the excess network bandwidth.

5. A Scheduling Algorithm for Downlink Traffics

The algorithm we are going to analyze now is a priority based scheduling algorithm to manage downlink traffics in IEEE 802.16 networks broadband wireless networks [4]. The objective of this algorithm is (i) to improve the network throughput, (ii) to satisfy delay constraints of real-time traffic and (iii) to achieve fair resource distribution among various MSS's (Mobile Subscriber Station). This algorithm arranges the resources in downlink traffic in an IEEE 802.16 broadband wireless network. The MSS's with good channel conditions will have a higher priority compared with those having bad channel conditions. Also MSS's with urgent real-time traffics will be assigned with a higher priority. Here starving is avoided by raising the priorities of non-real time traffics.

Given '*n*' MSS's where each of them requests an average real time data r_i^R and channel rate C_i. In this algorithm, the scheduler will first query the physical layers for the current channel rate C_i of each MSS and total free space '*F*' of the current downlink subframe. Then according to the rate requirement $(r_i^R, r_i^N(\text{Non-real time}))$ and buffered data (b_i^R, b_i^N) of each MSS, the scheduler will calculate its priority.

Based on these priorities, this algorithm is executed. For each MSS, its admitted real-time data rate r_i^R and non real-time data r_i^N are selected randomly for simulation process. A three-state Markov Chain model is used for the changes of the channel conditions of a MSS during simulation.

The comparison of the scheduling algorithm against the Max-Throughput (MT) scheme and the MPF scheme are shown in Fig. 4.

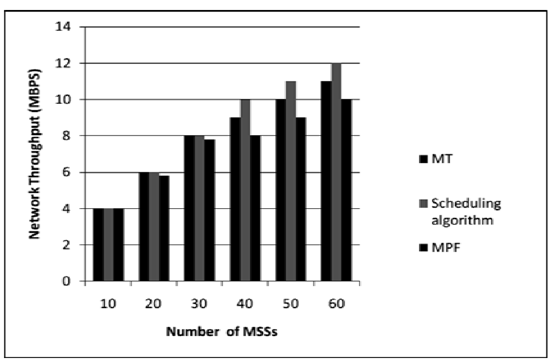

Figure 4

The simulated results of this algorithm has shown that this scheduler algorithm can improve the network throughput, maintain fairness by decreasing the packet dropping ratio of real-time traffics and increase the satisfaction ratios of non-real time traffics.

6. Conclusion

In this paper, we have analyzed priority queues and various algorithms using PQ models. In present scenario, PQ concepts play an important role in scheduling traffics in communication networks. Even though various queueing models are existing, priority queueing model is used to solve certain traffic conditions where all the data packets are not of equal importance or priority.

References

[1] D. Gross and C. M. Harris, Fundamentals of Queueing Theory, 2nd ed., John Wiley and Sons, New York, 1985.

[2] Aleksandar Kuzmanovic and Edward W. Knightly, TCP-LP: A distributed algorithm for low priority data transfer, Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, USA, IEEE INFOCOM, 2003.

[3] Cuixiang Kong, Zhonghua Guo, Liao Ping and Xuange Peng, PSRED: A queue management algorithm with Priority Self-adaptive Random Early Detection for Ad-hoc Network, Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009), Qingdao, China, November 21-22, 2009.

[4] Jia-Ming Liang, Jen-Jee Chen, You-Chiun Wang, Yu-Chee Tseng and Bao-Shuh P. Lin, Priority-based scheduling algorithm for downlink traffics in IEEE 802.16 networks. (http://people.cs.nctu.edu.tw/~yctseng/papers.pub/mobile05-apwcs-2009.pdf)

[5] Joris Walraevens, Bart Steyaert and Herwig Bruneel, Performance analysis of a single-server ATM queue with a priority scheduling, Computers & Operations Research 30 (2003), 1807-1829.