ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

COMPARATIVE ANALYSIS OF DIFFERENT QUEUING MECHANISMS IN HETROGENEOUS NETWORKS

Shubhangi Rastogi[1], Samir Srivastava[2]

M.Tech Student, Computer Science and Engineering, KNIT, Sultanpur, India[1]

Associate Professor, Computer Science and Engineering, KNIT, Sultanpur, India[2]

Abstract: Each router in the network must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. So Queuing is one of the important mechanisms in traffic management. In this paper various existing queuing algorithms are explained First in First out (FIFO), Priority Queue (PQ), Fair Queue (FQ), Weighted Fair Queue (WFQ), Class based WFQ (CBWFQ), Self Clocked FQ (SCFQ), Worst case fair WFQ (WF^2Q), Worst Case fair WFQ plus (WF^2Q+), Round Robin (RR), Weighted RR (WRR), Deficit WRR (DWRR). Then a comparative analysis of three queuing mechanisms FIFO, PQ and WFQ.WFQ technique has a superior quality than the other techniques.

Keywords: FIFO, PQ, FQ, WFQ, CBWFQ, DWRR.

I. INTRODUCTION

Congestion in Internet occurs when the link bandwidth exceeds the capacity of available routers or when the arrival rate of packets is greater than the departure rate due to one of the following two reasons: Input interface is faster than the output interface or Output interface is receiving packets coming in from multiple other interfaces. This results in long delay in data delivery and wasting of resources due to lost or dropped packets. The primary role of a router is to switch packets from the input links to output links through buffer. Apart from forwarding the packets, routers are involved for controlling the congestion in the network. It is known from that routing algorithms focus on two main concepts namely queue management and scheduling. Queue management algorithms manage the length of packet queues by dropping packets whenever necessary whereas scheduling algorithms determine which packets to be sent next. These algorithms are used primarily to manage the allocations of bandwidth among various flows. So objective of this paper is to analyze different existing queuing mechanisms and produces a comparative picture. Initial implementations of queuing used a single FIFO (first-in-first-out or first-come-first-serve queuing) strategy. More complex queuing mechanisms were introduced when special requirements need routers to differentiate among packets of different importance.

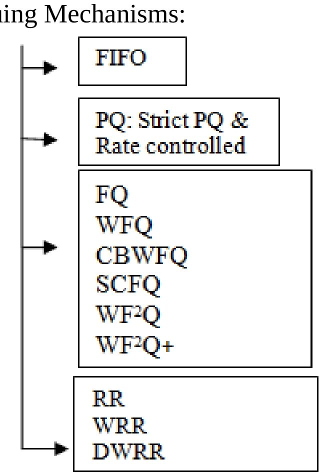
Queuing was split into two parts: first is the hardware queue that still uses FIFO strategy, which is necessary for the interface drivers to transmit packets one by one. The hardware queue is sometimes referred to as the transmit queue or TxQ and the second is the software queue that schedules packets into the hardware queue based on the QoS requirements.

Each queuing mechanism has three main components that define it:

1. Classification (selecting the class)
2. Insertion policy (determining whether a packet can be enquired)
3. Service policy (scheduling packets to be put into the hardware queue)

II. BACKGROUND

Existing Queuing Mechanisms:



ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

There are many more queuing algorithms but in this paper the main focus is on three algorithms i.e. FIFO, PQ and WFQ.

A. FIFO

First in First out i.e. FIFO is the most basic queuing mechanisms. It is simplest queuing mechanisms as in FIFO queuing there exist no classification because all packets are treated equally by placing them in a single class. Here Tail Drop policy is used in which packets are dropped when the output queue is full. The scheduler services packets in the order they arrived.

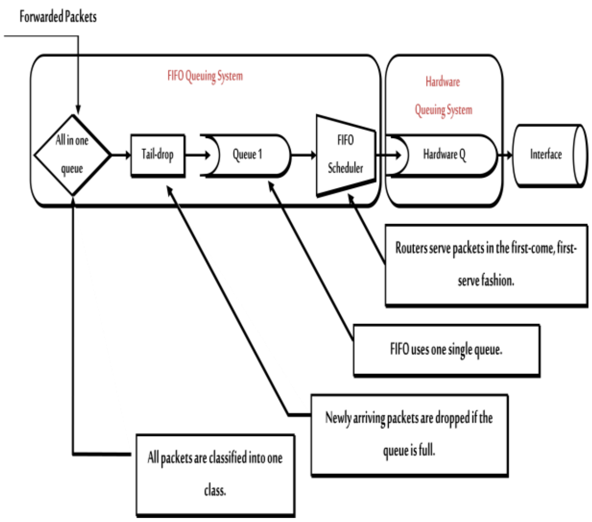


Fig. 1. FIFO Queuing

B. PQ

Priority Queue PQ is the basis for various queuing mechanisms. Packets are first classified into different priority queues as High queue, Medium queue, Normal queue (the default queue) and Low queue. Scheduling prefers packets in the same order. Each class uses one FIFO queue, where packets are dropped if a queue is full.

In Strict priority queue, first packets in High queue are scanned if there exist packets in high queue then these packets are scheduled first otherwise move to next queue and scan the medium and so on.

While in rate controlled Priority queue, higher priority packets are scheduled before lower priority one if the amount of traffic is below the user-configured threshold.

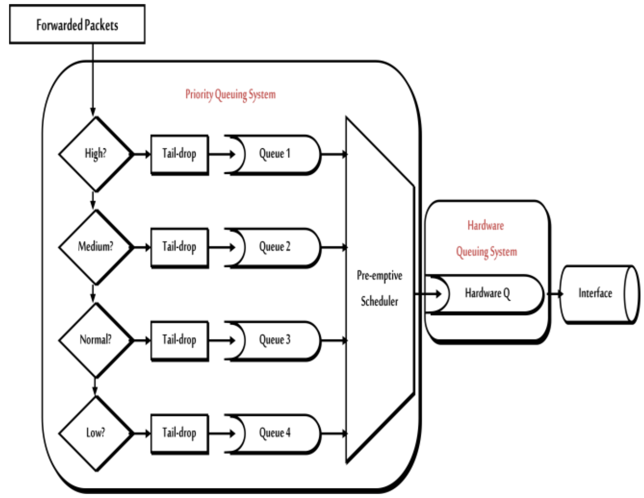


Fig. 2. Priority Queuing

C. FQ

Fair Queuing FQ is the foundation of various queuing mechanisms that are designed to ensure that each flow has fair access to network resources. Bandwidth is allocated fairly among all the flows. So packets arrived first (in FIFO) and packets of high priority will not monopolize the whole bandwidth. Here the flows which are sending more number of packets or large packets in size i.e. the packets of aggressive flows are dropped. So the misbehaving flow will not be destructive to other better behaved applications. Fair queuing algorithms can achieve fair bandwidth allocations by maintaining per-flow state and information.

If there are n numbers of active flows then each flow is allocated 1/n of the output bandwidth. As the no. of flows changes, the amount of bandwidth allocated to each of the queue also changes.

1) WFQ:

WFQ is basis for a class of queuing mechanisms that are designed to address limitation of FQ. WFQ supports Flow with different bandwidth requirements by giving each queue a weight that assigns it a different percentage of output bandwidth. WFQ also supports variable length packets, so that flows with large packet are not allocated more bandwidth than flows with smaller packets. But to support fair allocation when forwarding variable length packets adds complexity to queue mechanisms.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

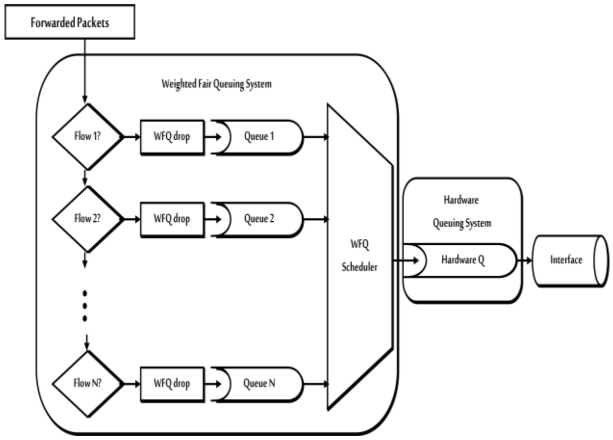


Fig. 3. WFQ Queuing

WFQ dropping is not a simple tail-drop. WFQ drops packets of the most aggressive flows.WFQ scheduler is a simulation of a TDM system (time-division multiplexer). The bandwidth is equally distributed to all active flows. Classification identifies a flow and assigns a queue to the flow.

WFQ Insertion and Drop Policy

WFQ has two modes of dropping:

1. Early dropping when the congestion discard threshold (CDT) is reached- to start dropping packets of the most aggressive flow, even before the hold-queue limit is reached

2. Aggressive dropping when the hold-queue limit (HQO) i.e. total maximum number of packets is reached

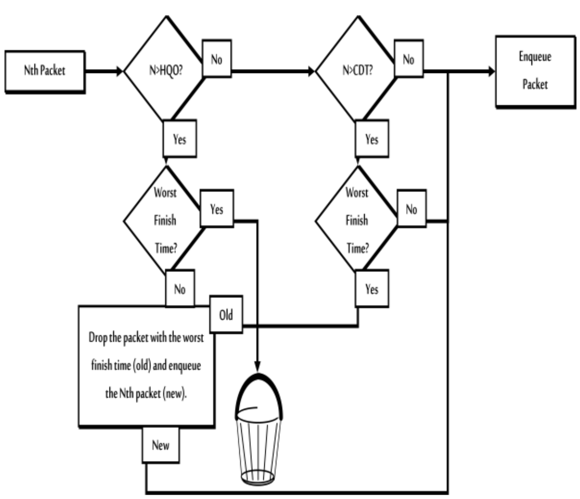


Fig. 4. WFQ insertion & dropping policy

The figure illustrates the dropping scheme of WFQ. The process can be split into the following steps:

Step1: Drop the new packet if the WFQ system is full (hold-queue limit reached) and the new packet has the worst finish time (the last in the entire system).

Step2: Drop the packet with the worst finish time in the WFQ system if the system is full and enqueue the new packet.

Step3: Drop the new packet if the queue, where the packet should be enqueued, is the longest (not in packets but in the finish time of the new packet) and there are more packets in the WFQ system than the CDT.

Step4: Otherwise enqueue the new packet.

2) CBWFQ:

Class-based weighted fair queuing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class. A queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class. Once a class has been defined according to its match criteria, you can assign it characteristics. To characterize a class, you assign it bandwidth, weight, and maximum packet limit. The bandwidth assigned to a class is the guaranteed bandwidth delivered to the class during congestion. To characterize a class, you also specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in the queue for the class. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class. After a queue has reached its configured queue limit, enqueuing of additional packets to the class causes tail drop or packet drop to take effect, depending on how class policy is configured. Once a packet is classified, all of the standard mechanisms that can be used to differentiate service among the classes apply. Flow classification is standard WFQ treatment. That is, packets with the same source IP address, destination IP address, source Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port, or destination TCP or UDP port are classified as belonging to the same flow. WFQ allocates an equal share of bandwidth to each flow. Flow-based WFQ is also called fair queuing because all flows are equally weighted. For CBWFQ, which extends the standard WFQ fair queuing, the weight specified for the class becomes the weight of each packet that meets the match criteria of the class. Packets that arrive at the output interface are classified according to the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

match criteria filters you define, and then each one is assigned the appropriate weight. The weight for a packet belonging to a specific class is derived from the bandwidth you assigned to the class when you configured it; in this sense the weight for a class is user-configurable. After the weight for a packet is assigned, the packet is enqueued in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the class queue is serviced fairly.

3) SCFQ:

The Self-Clocked Fair Queuing, SCFQ is enhancement to WFQ that simplifies the complexity of calculating the finish time in a corresponding GPS system. The decrease in complexity results in a larger worse-case delay and delay increases with the number of service classes. However, it was discovered that calculating the Finish Time (the time at which a packet would have been serviced given a hypothetical fluid server) for WFQ was complicated/difficult. SCFQ uses a simplified method of calculating the service time, based on the transmission delay of the packet, and the finish time of the packet currently being serviced.

4) WF^{2}Q:

The Worst-case Fair Weighted Fair Queue is an enhancement to WFQ that uses both the start time and finish times of packets to achieve a more simulation of a GPS system. It is neither forward nor back more than one package length. For avoiding oscillations between high and low services for a flow. WF^{2}Q is also more suitable for the resumption of the congestion and control algorithms. In a system that uses WF^{2}Q, the server selects the following packages which must be processed among those packages the GPS system would have sent.
Among these packages it is chosen that one who has minimum reference time in GPS. WF^{2}Q is a better packet approximation algorithm of GPS than WFQ. It provides almost identical service with GPS, the maximum difference is no more than one packet size. The problem with WF^{2}Q is the time complexity for computing the virtual time.

5) WF^{2}Q+

The Worst-case Fair Weighted Fair Queue + is an enhancement of WF^{2}Q which implements a new virtual time function that results in lower complexity and higher accuracy. Scheduler maintains a virtual clock in addition to the real clock and separate queue for each session. When a packet reaches the head of its queue, it is assigned a Virtual Start Time and a Virtual Finish Time. Only the packets at the head of their queues with Virtual Start Time less than or equal to the current Virtual Time are eligible for transmission. Among eligible packets, the one with the least Virtual Finish Time is picked for transmission.

III. COMPARATIVE ANALYSES OF QUEUING TECHNIQUES

S. No.	Queuing Algorithm	Description	Advantages	Limitations	Applications
1	FIFO	-no. of queues= 1 -based on first come first served -packet by packet dispatching -tail drop mechanism used -enabled when bandwidth is more than 2 Mbps -high processing speed	-simple and fast -no need to configure -low computational load -predictable in nature -no reordering -supported on all platforms	-unfair bandwidth allocation -causes starvation -causes jitter -do not allow routers to organize buffer packets	gives benefits to UDP flows over TCP flows and gives acceptable results for FTP
2	PQ	-no. of queues= 4 -high priority packets are serviced first -packet by packet dispatching - tail drop mechanism used -designed for low bandwidth links -low processing speed	-low delay to high priority packets -low computational load - allow routers to organize buffer packets	- unfair bandwidth allocation -starvation of lower priority packets -need to be configured -low processing speed	Used in real time applications as VOIP

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

3	WFQ	-number of queues are configurable -low volume traffic is given priority -conversational dispatching -Modified tail drop mechanism(drops most aggressive flow) - enabled when bandwidth is less than 2 Mbps - processing speed is faster than PQ but is slower than FIFO	-easy to configure -fair bandwidth allocation -reduced jitter -proportional bandwidth for traffic of different priorities	-complex -not applicable to delay sensitive real time services	Works best for FTP and video conferencing

IV.CONCLUSION

After surveying the various existing queuing mechanisms and comparatively analyzing the queuing mechanisms FIFO, PQ and WFQ we conclude that WFQ performs the best among above mentioned mechanisms in most of the applications. WFQ is not suitable for delay sensitive traffic such as voice in VOIP application. PQ gives the best result for delay sensitive data so it is suitable for VOIP. Whereas FIFO is simple and fast queuing mechanism in which there is no need of reordering and configuring the packets. So WFQ gives good performance in FTP, Video conferencing and many more applications.

REFERENCES

[1] G.Thiruchelvi and J.Raja, "A Survey On Active Queue Management Mechanisms", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008.
[2] Jung-Shim Li, Yi-Ting Lin, Ching-Fang Yang, "CORE-STATELESS FAIR RATE ESTIMATION FAIR QUEUING" 0-7803-7625-0/02/$17.0W0O 0 2 IEEE British Crown Copyright.
[3] Mohammad Mirza Golam Rashed and Mamun Kabir, "A COMPARATIVE STUDY OF DIFFERENT QUEUING TECHNIQUES IN VOIP, VIDEO CONFERENCING AND FILE TRANSFER", Daffodil International University Journal of Science and Technology, volume 5, issue 1, January 2010.
[4] Anca VARGATU, Ionut BOSOANCA, "A Survey of Wireless Fair Queuing Algorithms with Location-Dependent Channel Errors" Informatica Economica vol. 15, no. 4/2011.
[5] Tomas Balogh, Martin Medvecký, "Comparison of Priority Queuing Based Scheduling Algorithms" VOL. 1, NO. 1, APRIL 2010
[6] QoS Congestion Management (queuing) [ONLINE] Available at: http://www.cisco.com/en/US/tech/tk543/tk544/tsd_technology_support_protocol_home.html
[7] P. Ramanathan and P. Agrawal," Adapting packet fair queuing algorithms to wireless networks", Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking, pp.1-9, 1998

BIOGRAPHY

Shubhangi Rastogi has completed her B.Tech from Shri Ram Murti Smarak College of Engineering and Technology, Bareilly (U.P.) in 2011 with Computer Science and Engineering stream. Currently she is pursuing M.Tech from Kamla Nehru Institute of Technology, Sultanpur with Computer Science and Engineering stream. Her areas of interest are Computer Networks, Cryptography and Parallel Algorithm.