Active Scheduling (Queuing) Algorithms in Congestion Management: A Review

1Nidhi Malhotra, 2Anil Kumar Sharma
1M. Tech Scholar, Deptt. of ECE, Institute of Engineering & Technology, Alwar, Raj.-301030 (India)
E-mail: ece.nidhi@gmail.com
2Professor & Vice Principal, Institute of Engineering & Technology, Alwar, Raj.-301030 (India)
E-mail: aks_826@yahoo.co.in

Abstract— Active Queue Management (AQM) algorithms have been designed to be able to actively control the average queue length in routers supporting TCP traffic, and thus to be able to prevent congestion and resulting packet loss as much as possible. Many different algorithms have already been presented in literature, of which all algorithms have been given the most attention. In order to be able to compare these algorithms, various approaches can be applied to model the algorithm behaviour, and subsequently to derive interesting performance measures. These modelling approaches have certain advantages over algorithm simulation, which has mostly been applied to derive algorithm performance measures up to now; here we take a review over the available queuing techniques in congestion management.

Keywords— Congestion Management, Scheduling Problem QoS, Queue Management.

I. INTRODUCTION
The amount of traffic carried over the Internet is dramatically increasing. There is no doubt that the Internet is becoming a public communication infrastructure. With the rapid growth of the Internet, customers are demanding multimedia applications to be available on the Internet. There is a huge demand for the Internet as the number of services keeps increasing. However, some essential features must be provided if it is to be accepted more widely. One is to achieve fair bandwidth allocation among competing flows. Router mechanisms designed to achieve fair bandwidth allocations, like Fair Queuing have many desirable properties for congestion control in the Internet. However, such mechanisms usually need to maintain state, manage buffers, and/or perform packet scheduling on a per flow basis, and this complexity may prevent them from being cost effectively implemented and widely deployed. As part of the resource allocation mechanisms, each router must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. Various queuing disciplines can be used to control which packets get transmitted (bandwidth allocation) and which packets get dropped (buffer space). The queuing discipline also affects the latency experienced by a packet, by determining how long a packet waits to be transmitted. Examples of the common queuing disciplines are first-in first-out (FIFO) queuing, priority queuing (PQ), and weighted-fair queuing (WFQ). In order to complete various processes successfully, the network should maintain a good QoS (Quality of Service) to provide satisfactory results to the user. QoS must be efficient to differentiate the traffic and satisfy their specific requirements. As the traffic on network is increasing due to congestion, it decades the performance of network, the different priorities can be assigned to different applications to enhance the performance of network. The affects of different queuing disciplines on packet delivery for various applications has been observed. The queuing disciplines are used to control which packets get transmitted (bandwidth allocation) and which packets get dropped (buffer space). The main function of the queuing discipline is to control how packets enqueued on that particular link. In scheduling data packets over the network, queue service disciplines are used to determine service priority, delay bound, jitter bound and bandwidth. The various queuing disciplines that are used to transmit packets are: FIFO (First In first out), PQ (Priority queue), WFQ (Weighted Fair Queue), CQ (Custom Queue), MDRR (Modified Deficit Weighted Round Robin), MWRR (modified weighted Round robin), DWRR (Deficit Weighted Round robin), FRED (Flow random Early Drop) and CSFQ (Core Stateless fair Queuing). There are several tasks that any scheduling discipline should accomplish:

- Support the fair distribution of bandwidth to each of the service classes competing for bandwidth on the output port.
- Furnish protection (firewalls) between the different service classes on an output port so that a poor service class queue cannot
impact in bandwidth & delay delivered to other service classes assigned to other queues on the same port.

- Allow other service classes to access bandwidth that is assigned to a given service class.
- Provide an algorithm that can be implemented in hardware, so it can arbitrate access to bandwidth on the higher-speed router interfaces without negatively impacting system forwarding performance.

QoS (Quality of Service)

Quality of Service (QoS) is the quality a user or customer can expect from a given service. Also QoS is defined as the proficiency of a network element to furnish some degree of commitment for congenial network data delivery. In other words, QoS means, satisfying customer application requirements, providing a network that is transparent to its users. QoS does not generate bandwidth.

When specifying the QoS, a number of factors are taken into account:

- **Latency** - the time from a packet is sent until it is received at another point. Response time is another term concerning latency, and refers to the round-trip time, i.e. twice the latency. For IP telephony, this is a very important factor.
- **Jitter** (timing jitter) – timing variations from an ideal position in time, caused by packets arriving either out of order or at an inconsistent rate. This is particularly damaging in multimedia applications where timing inconsistencies of the data may be viewable as shaky images in real-time video applications.
- **Packet Loss** - the percentage of packets lost in the transmission. Different applications will have different tolerance of packet loss.
- **Throughput** - the amount of data transferred between two given nodes during a given amount of time. This reflects the bandwidth of the network and is a significant factor to QoS for e.g. videoconferences.

Quantifying the above parameters allows us to find out how efficiently the traffic in an IP network is being managed and whether the network is suitable for the data we wish to transmit or not. Different kinds of applications have different requirements for the parameters listed above [3].

II. AN OVERVIEW OF EXISTING AQM ALGORITHMS

This chapter attempts to give a representative overview of existing active queue management (AQM) algorithms, as they have been presented in the literature. Since many different algorithms have been proposed, and many of these only differ in minor details, this overview does not pretend to be complete, but is limited to the most important algorithms and algorithm classes. First, the general purpose of AQM algorithms is presented. Then, a classification of the algorithms is given, in order to provide a clear structure for the algorithms mentioned in the rest of this chapter. For each algorithm class, the most significant algorithms belonging to this class are then presented in some detail. Queuing can be divided into four basic activities according to:

- Adding a packet to the correct queue. The queue to add the packet to is determined during the classification stage.
- Dropping a packet if the queue becomes full.
- Removing a packet if requested by the scheduler.
- Optionally monitor the status of the queue(s) and take proactive steps (removing or marking packets) to keep the occupancy level low [4].

There are several algorithms for traffic management that actively control the average queue length in traffic routers, some important of them are described below:

**First-in, First-out Queuing (FIFO)**

FIFO is a simple tail-drop queuing mechanism. This is the simplest and most common interface queuing technique and works well if links are not congested. The first packet to be placed on the output interface queue is the first packet to leave the interface, as explained in figure 1.

![Fig 1.1 Impact of file transfer on interactive traffic with FIFO](image)

The problem with FIFO queuing is that when a host starts a file transfer, it can consume all the bandwidth of a link to the disadvantage of real-time traffic. The phenomenon is referred to as packet trains because one source sends a “train” of packets.
to its destination and packets from other hosts get caught behind the train. First in, first out queuing is efficient for large links that have little delay and minimal congestion.

**Priority Queue (PQ)**

PQ ensures that important traffic gets the fastest handling at each point where it is used. It is designed to give strict priority to important traffic according to what is set by the classifier. In PQ, each packet is placed in one of four queues: High, Medium, Normal, or Low based on an assigned priority. Packets that are not classified by this priority-list mechanism fall into the Normal queue. During transmission, the algorithm gives higher-priority queues absolute preferential treatment over low-priority queues.

![Fig1.2 Priority Queuing](image1.png)

**Weighted Fair Queuing (WFQ)**

WFQ is the best known and the most studied queuing discipline. It assigns a queue for each flow and applies priority (or weights) to identified traffic to determine how much bandwidth each flow is allowed relative to others. As a result, WFQ can prevent other flows to have direct influence on one specific flow. Flows are broken into two categories: those requiring large amounts of bandwidth and those requiring a relatively small amount of bandwidth. The goal is to always have bandwidth available for the low throughput flows and allow the high throughput flows to split the rest proportionally to their weights [5].

![Fig1.3 Fair Queuing](image2.png)

**Custom Queuing (CQ)**

Custom Queuing shown in Figure 1.5 allows users (customers) to reserve a percentage of bandwidth for specified protocols. The users can define up to 10 output queues for normal data and an additional queue system for system messages such as LAN keep alive messages. In CQ routers service each queue sequentially transmitting a configurable percentage of traffic on each queue before moving on to the next one.

![Fig1.4 Weighted Fair Queuing](image3.png)
CQ guarantees that mission critical data is always assigned a certain percentage of bandwidth and also assures predictable throughout for other traffic. To provide these features routers determine how many bytes should be transmitted for each queue based on the interface speed and the configured percentage. When a particular queue is being processed, packets are sent until the number of sent bytes exceeds the Byte Count, or until the queue is empty [6].

**Deficit Weighted Round Robin (DWRR)**

Deficit Round Robin (DRR) addresses the limitation of WFQ and WRR. It is a modification of WRR. The improvement to this algorithm is that it can handle variable packet sizes without knowing the average packet size of the flows. DRR services the queues in a round robin order. Each queue is allowed to send a certain amount of bytes in each round.

There are two variables associated with each queue in this algorithm: Quantum and Deficit Counter. Quantum represents the number of bytes that each queue can send on its turn. The Deficit Counter variable is used to keep track of the credit each queue possesses for sending traffic and is initialized to zero.

**Modified Weighted Round Robin (MWRR)**

Modified WRR is a best effort connection scheduling discipline. It’s the simplest emulation of generalized processor sharing discipline. While GPS serves infinitesimal amount of data from each non empty connection MWRR serves a number of packets for each non empty connection where

\[
\text{Number} = \text{normalized} \left( \frac{\text{mean packet size}}{\text{Weight}} \right)
\]

To emulate GPS correctly when packets can be of different sizes, a weighted round robin server must know a source’s mean packet size in advance.

**RED (random early drop)**

The basic idea behind RED queue management is to detect incipient congestion early and to convey congestion notification to the end-hosts, allowing them to reduce their transmission rates before queues in the network overflow and packets are dropped. To do this, RED maintains an exponentially-weighted moving average (EWMA) of the queue length which it uses to detect congestion. When the average queue length exceeds a minimum threshold (min), packets are randomly dropped or marked with an explicit congestion notification (ECN) bit. When the average queue length exceeds a maximum threshold (max), all packets are dropped or marked. Random Early Detection (RED) keeps no per flow state information. Packets are dropped probabilistically based on the long-term average queue size and fixed indicators of congestion (thresholds).

**Flow Random early drop (FRED)**

Flow Random Early Drop (FRED) is a modified version of RED, which uses per-active-flow accounting to make different dropping decisions for connections with different bandwidth usages. FRED only keeps track of flows that have packets in the buffer, thus the cost of FRED is proportional to the buffer size and independent of the total flow.
numbers (including the short-lived and idle flows). FRED can achieve the benefits of per-flow queuing and round-robin scheduling with substantially less complexity. Flow Random Early Drop (FRED) uses per-flow preferential dropping to achieve fairer allocation of bandwidth among flows. FRED builds per-flow state at the router by examining those packets that are currently in the queue. The packet drop rate for a flow is determined by the number of packets the flow has in the queue, and is not directly influenced by the flow’s data rate or round trip time. We evaluate the effectiveness of FRED as a less expensive means of attempting per-flow fairness.

Core Stateless Fair Queuing (CSFQ)
The Core-Stateless FQ scheme, CSFQ, distinguishes core routers, the higher-speed and busier routers at the “core” of an Internet AS backbone from edge routers. In typical deployment, edge routers might handle thousands of flows, while core routers might handle 50k-100k flows. CSFQ exploits this gap by delegating the management of per-flow statistics to the edge routers. Edge routers then share this information with core routers by labelling each packet that they forward. Core routers, in turn, can use the labels to allocate bandwidth fairly among all incoming flows. It is important to realize that in the case of CSFQ, edge routers run essentially the same algorithm as core routers (including probabilistically dropping incoming packets); however, edge routers have the added responsibility of maintaining per-flow state. In general, of course, edge and core routers in such an approach could run very different algorithms.

III. AQM MODELLING APPROACHES
This section (Literature survey) of the paper describes some of the work simulation on the techniques described above, to provide a better congestion control mechanism in existing scheduling environment.

Velmurugan et al. (2009) described a simulation environment, in which various resource allocation mechanisms were studied. In this paper, queuing disciplines were implemented on some routers that showed that how packets were buffered while waiting to be transmitted. Examples of the common queuing disciplines used were first-in first-out (FIFO) queuing, priority queuing (PQ), and weighted-fair queuing (WFQ). It had also thrown a light on the comparative study of these disciplines and explained through simulation which was a better method of queuing.

Lee et al. (2007) presented a scheme called Extended Deficit Round Robin that was a combination of hop-by-hop credit based flow control with a modified version of Deficit Round Robin scheduling to achieve network wide max-min fair share. Deficit Round Robin scheduling had been proposed in to alleviate the problem of unequal sharing. EDRR was the first scheme that combines DRR with hop-by-hop credit based flow control [7].

Subhash et al. (2006) discussed the work performance of scheduling disciplines like FIFO, PQ, WFQ, MWRR, CQ and DWRR in packet dropping and reception for a high speed network are simulated. OPNET IT GURU a network simulator from OPNET Technologies has been used for simulation. It has been observed that PQ, WFQ, MWRR, DWRR perform alike in dropping packets. The bursty nature of WFQ didn’t make it to receive any voice traffic over all intervals of time. Regarding video conferencing traffic a sharp fall in received data was observed for PQ and CQ [8].

Sulaiman et al. (2006) presented an analytical study on Voice over Internet Protocol (VoIP) performance in a third generation (3G) network evaluated under certain IP conditions. The delay was analysed using different types of coding schemes to estimate the actual voice packet arrival as traffic sources are very bursty in nature. A model was simulated for different value of packet arrival rate and different types of encoder. The simulation results match to the theoretical model and it also indicated that encoder types have an impact on the system performance. The stability of a system was also affected by the rate of voice traffic [9].

Salami et al. (2006) discussed the two basic queuing strategies in routers are Output Queuing (OQ) and Input Queuing (IQ). OQ guarantees QoS and gives optimal throughput but it was not scalable. This paper proposed a Multistage Queuing and Scheduling strategy in which VOQ (virtual output queue) was implemented at the input ports and OQ at the output ports of the router. The scheduling algorithm for the VOQ presented in this paper was an Iterative Probabilistic Scheduling [10].
As the volume of traffic and number of simultaneously active flows on Internet backbones increases, the problem of recognizing and addressing congestion within the network becomes increasingly important. There are two major ways in which routers can help in congestion control: by signalling congestion using either packet drops, and by cleverly managing its buffers. Today ‘Scheduling’ is the third way in which routers can assist in congestion control. A particular form of scheduling, called fair queuing (FQ) achieves (weighted) fair allocation of bandwidth between flows. While FQ achieves weighted-fair bandwidth allocations, it requires per-flow queuing and per-flow state in each router in the network. This is often prohibitively expensive. Another scheme, called core-stateless fair queuing (CSFQ), displays an interesting design where “edge” routers maintain per-flow state whereas “core” routers do not. They use their own non per-flow state complimented with some information reaching them in packet headers (placed by the edge routers) to implement fair allocations. The use of a queuing network model helps in extending the traffic model to incorporate non-homogenous traffic. Over the years simple product form results have been published for many queuing networks. There are many more network models which include general service and arrival patterns. Thus, with help of actual road data, various combinations of routing probabilities between nodes can be used to model different vehicle types. This paper only gives a flavour of a fresh analytic approach towards modelling congestion. Also, incorporating more well-known queuing network models and onsite data, a more exact Road Cell Network model can be built. Such models, apart from quick approximations, can also help in quicker simulations using perturbation analysis. Literature survey that has been done to find the available gap in the literature and to define the problem. We identified different inputs and outputs for the present system and analysed the performance and capability of some important queuing techniques. How to improve the QoS of network using queuing disciplines or by using congestion management is discussed and it concludes that how much work is done till now by different people in this research area.

REFERENCES


High Speed Networks” IEEE/ACM transactions on Networking, Vancouver, pp. 33-46.


Authors Profile

Nidhi Malhotra has completed his bachelor degree from MDU, Rohtak in 2005. She is currently doing his M Tech thesis work at the Dept. of ECE, Institute of Engineering & Technology, Alwar affiliated to Rajasthan Technical University, Kota. Her research area is WSN, Microprocessor, Digital communication.

Dr. Anil Kumar Sharma MIEEE received his M.E. degree in Electronics & Communication Engineering (ECE) from Birla Institute of Technology, Deemed University, Mesra, Ranchi, India, in 2007 with CGPA of 8.45 and Ph. D in ECE in 2011. He has an experience of 20 years on various RADARs and Communication Equipments. He is currently a Professor in the Department of ECE and Vice Principal, Institute of Engineering and Technology, Alwar- 301030, Rajasthan, India. He has published 17 papers in International Journals, over 40 papers in various Conferences and authored 04 books. His research/teaching interest includes WSN, VLSI Design, RADARs and Soft Computing.