Comparative Analysis of Queue Scheduling Policies

Er. Palak Batra
M.Tech (Student), CSE Section, JCD College of Engg. (Sirsa)

Er. Amit Nagpal
M.Tech (Student), ECE Section, JCD College of Engg. (Sirsa)

Abstract— As the growth of network application and internetworking is rising exponentially, numerous requirements are also emerging on the screen. Supporting current and multimedia applications requires the network to ensure the acceptable level of service to packet flow over shared and scarce resource. To attain the QoS (quality of service) and to avoid the congestion, the most common and cheapest approach is scheduling mechanism and packet discarding policies. Here, in this paper, the detailed comparative study has been done on various queue scheduling mechanisms and analytic results has been given for the two randomly chosen mechanisms—Drop tail(DT) and RED(random early detection). The analytic comparison has been done on considering the scenarios like, analyzing the performance of DT and RED at different level of congestion and considering variable queue size.

Keywords: QoS, TCP, Scheduling Algorithms.

I. INTRODUCTION

The goal of queue scheduling is to achieve high link utilization without introducing excessive delays into the end to end path. For good link utilization, it is necessary that the queue can accommodate for variations in traffic load. These variations include both sudden packet bursts, as well as load changes due to the congestion control mechanisms of TCP. The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite. TCP provides reliable, in-order delivery of a stream of bytes, making it suitable for applications like file transfer and e-mail. It is so important in the Internet protocol suite that sometimes the entire suite is referred to as "the TCP/IP protocol suite." In TCP, each end host controls its packet transmission rate by changing the window size in response to network congestion. A key is that the TCP congestion control is performed in a distributed fashion; each end host determines its window size by itself according to the information obtained from the network. In general, there are two major objectives in the congestion control mechanism. The one is to avoid an occurrence of the network congestion, and to dissolve the congestion if the congestion occurrence cannot be avoided. The other is to provide fair service to connections. Keeping the fairness among multiple homogeneous/heterogeneous connections in the network is an essential feature for the network to be widely

accepted. The fair service also involves detecting misbehaving flows, which do not properly react against the network congestion and unfairly occupy the network resources (such as router buffer and link bandwidth) [9].

II. QUEUE SCHEDULING

As the growth of network applications and internetworking is rising exponentially, numerous requirements are also emerging on the screen. Supporting current and emerging multimedia applications requires the network to ensure acceptable level of service to packet flows over shared and scarce resources (e.g. link bandwidth and buffer space). For example, real-time traffic such as interactive voice cannot tolerate delay of more than 250ms. File transfer, on the other hand, is more sensitive to packet loss but can tolerate delay. Compressed video is more sensitive to delay variation or jitter but is flexible in terms of packet loss (up to 1% is acceptable without severe degradation of quality). Several mechanisms have been proposed and deployed in today's network devices to support required QoS such as resource reservation, admission control, traffic shaping, scheduling and packet discarding policies. The cheapest and the most common approach are scheduling mechanisms and packet discarding policies. The goal of queue scheduling is to achieve high link utilization without introducing excessive delays into the end to end path. For good link utilization, it is necessary that the queue can accommodate for variations in traffic load. These variations include both sudden packet bursts, as well as load changes due to the congestion control mechanisms of TCP. Basically, Queue management is the decision when to start dropping packets and which packets to drop at a congested router output port [1]. Here is the list of some flows and their acceptable latency and their drop probability.

Sr. No.	Services	Expected Bandwidth Shared	Maximum Desired latency	Desired Packet Length	Drop Probability
1	VoIP	3%	20 ms	128	N/A
2	DNS	2%	50 ms	256	N/A
3	FTP	20%	100 ms	1024	4%
4	Mail	10%	100 ms	1024	4%
5	WWW	8%	100 ms	512	1%

Table 1. Desired Specifications for the different Services

III. SCHEDULING PARAMETERS

- Quality of service (QoS)**:** Scheduling algorithm should maintain QoS under normal conditions as well as under network degradation scenario.
- Data throughput and channel utilization: Scheduling algorithm should utilize the channel properly and avoids wastage of bandwidth.
- Fairness: It refers to optimizing the capacity of the channel by giving preference to spectrally efficient modulations while still allowing transmissions with more robust modulations.
- Power constrain: Scheduling algorithm should consume little power.
- Simplicity: Scheduling algorithm should be not being too complex as there are computational limitation at both BS and MS.

IV. SCHEDULING ALGORITHMS

There are many algorithms that provide scheduling. Each algorithm has its own characteristics, advantages and disadvantages as compared to other algorithms. Some scheduling algorithms provide balance throughput and fairness, some are good in terms that consume less power and are simple. But our main concern in selecting a scheduling algorithm that it should maintain QoS under normal conditions as well as under network degradation scenario.
Now we discuss some of the main scheduling algorithms in detail:

First-In-First-Out

This queueing is the most popular queue scheduling discipline that has been extensively examined in the literature. In FIFO queueing, all packets placed into a single queue and then served in the same order on which they arrived. Although it is simple and has predictable behavior and extremely low computational load on the system, it has some severe limitations for multimedia traffic. It is incapable of providing differentiated service and cannot isolate the effect of ill-behaved flow on other flows [5].

Priority Queuing

It is a simple approach to provide differentiated services to different packet flows. Packets of different flows are assigned a priority level according to their QoS requirements. When packets arrive at the output link, they are first classified into different classes enqueued separately based on their priorities. Then, queues are served in order. The highest priority queue is served first before serving lower priority queues. Packets in the same priority class are serviced in a FIFO manner. The limitation of PQ is that lower-priority packets may receive little attention when a higher-priority class has a continuous stream of packets. This problem is known as starvation problem. Also it lacks fairness [5][12].

Drop Tail

Drop Tail or Tail Drop, is a simple queue management algorithm used by Internet routers to decide when to drop packets. In contrast to the more complex algorithms like RED and WRED, in Tail Drop all the traffic is not differentiated. Each packet is treated identically. With tail drop, when the queue is filled to its maximum capacity, the newly arriving packets are dropped until the queue has enough room to accept incoming traffic. The name arises from the effect of the policy on incoming datagrams. Once a queue has been filled, the router begins discarding all additional datagrams, thus dropping the tail of the sequence of datagram's. The loss of datagrams causes the TCP sender to enter slow-start, which reduces throughput in that TCP session until the sender begins to receive ACKs again and increases its congestion window. A more severe problem occurs when datagrams from multiple TCP connections are dropped, causing global synchronization, i.e., all of the involved TCP senders enter slow-start. This happens because, instead of discarding many segments from one connection, the router would tend to discard one segment from each connection [10][12].

Fair Queuing

In FQ, the buffer space is divided into many queues to hold the packets destined for or from users. In order to decide which packet should be forwarded first, FQ estimates a "virtual" finishing time. Finally, FQ compares the virtual finishing time and selects the minimum one. The packet with the minimum "virtual" finishing time is forwarded [12].

Round Robin (RR) Scheduling

This is one of the simplest scheduling algorithms designed specially for a time sharing system, in which the scheduler assigns time slots to each queue in equal portions without priority. This algorithm is used for fair allocation of resources. This algorithm is called round robin as it works in the form of rounds, in each round every connection is served only once in there time quantum. It has the same bandwidth efficiency as a random scheduler. Also, it cannot guarantee different QoS requirements for each queue. An improvement in the RR scheduling algorithm is Weighted Round Robin (WRR). In this scheduling algorithm packets are first classified into various service classes and then assigned a queue that can be assigned a different percentage of bandwidth and is serviced in round robin order.[5]

Random Early Detection (RED)

Random Early Detection (RED) is produced by Internet Research Task Force (IRTF). Once a link is filling up when TCP/IP session starts, RED starts dropping packets with probability which indicate to TCP/IP that the link is congested and it should slow down. Once the link is completely saturated, it behaves like a normal traffic police. Use of dynamic drop probability ensures the gateway reacts differently to different level of congestion anticipation i.e. if queue-size is approaching thresholds the drop probability has to be higher than, say when the queue size is very less compared to the threshold.

Due bursty nature of Internet traffic, the instantaneous queue sizes are not a true indicator of congestion. Hence RED does not use instantaneous queue sizes; instead it uses average queue size measured over all times. The RED gateway calculates the average queue size as exponentially weighted moving average of instantaneous queue size [4].

V. ANALYTIC COMPARISON

Here, in this section, we have analyzed the performance of Drop Tail and RED through network simulator, NS-2. The first comparison is made by taking the scenario of level of congestion (Table2). As we increase the no. of TCP connections or level of congestion, the fairness index in case of Drop Tail reduces to an extent which means unfair use of connection bandwidth, whereas in case of RED, it also decreases, but does not create the unfairness.

Level of Congestion	Drop Tail	RED

2	0.96	0.97
4	0.99	0.99
8	0.98	0.99
16	0.63	0.82

Table 2. Fairness Index in case of increasing level of congestion

The second comparison is made on taking the variable queue size. As we increases the queue size, the fairness index in both the cases changes drastically (Table 3).

Queue Size	Drop Tail	RED
10	0.63	0.82
20	0.89	0.95
30	0.86	0.98

Table 3. Fairness Index in case of variable queue size

Still, RED has better performance. Because, with Drop Tail gateways with a small queue, the queue drops more packet while the TCP connection is in slow-start phase of rapidly increasing its window in case of RED. With the Drop Tail gateways, with the large maximum queue the average delay is unacceptably large. In addition, Drop Tail Gateways are more likely to drop packets from both the connections at the same time, resulting in global synchronization and further loss of throughput.

VI. CONCLUSION

Random Early Detection is an effective mechanism to avoid congestion. RED gateway drop packets when the average queue size exceeds the maximum threshold. The probability that the RED gateway chooses a particular connection to notify during congestion is roughly proportional to that connection's share of bandwidth at the gateway. This approach avoids a bias against bursty traffic. For RED gateways, the rate at which the gateway marks packets depends on the level of congestion, avoiding global synchronization that results from many connections decreasing their window at the same time. As in case of variable queue size, RED again outperforms. With Drop Tail gateways with a small queue, the queue drops packets while the TCP connection is in the slow start phase of rapidly increasing its window in case of RED. On the other hand, with Drop Tail gateways with the large maximum queue, the average delay is unacceptably large. In addition, the drop Tail gateways are more likely to drop packets from all the connections at the same time, resulting in global synchronization.

REFERENCES

[1] Christof Brandauer , " Comparison of Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet Traffic" International Conference on Advanced Communication Technology,pp.122-129,2001.

[2] Cheng-Chih Yang, " QoS Performance Improvement for WLAN using Priority Random Early Detection" IEEE, International Journal of Computer Science Issues IEEE, Vol. 2, 2007.

[3] Executive Summary Download: http://eeweb.poly.edu/aatcn/pres_reps/JTao_RED_report.pdf

[4] Floyd, S. & Jacobson, V, "Random Early Detection Gateways for Congestion Avoidance" IEEE/ACM Transactions on Networking. Vol. 3. 397-413, 1993

[5] Hafiz M.Asif and El-Sayed M.El-Alfy,"Simulation-Based Performance Comparison of Queueing Disciplines for Differentiated Services Using OPNET " pp 1-4

[6] Liu Enhai, Liu Yan, Pan Ruimin, "An Improved Random Early Detection Algorithm Based on Flow Prediction" Second International Conference on Intelligent Networks and Intelligent Systems, 2009.

[7] Mark Wurtzler, "Analysis and simulation of Weighted Random Early Detection", pp 5, 2002

[8] Mats Sagforsa, "Queue Management for TCP Traffic over 3G Links" IEEE pp. 1663- 1668, 2003.

[9] Md Tarmizil and A. Albagul, "QoS Evaluation of Different TCPs Congestion Control Algorithm Using NS2", vol. 6, 2006.

[10] Nemanja Vucevic and Jordi Perez-Romero, " Reinforcement learning for active queue management in mobile ALL-IP Networks" The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,2007

[11] SHU-GANG LIU, " Simulation and Evaluation of RED in congestion control" Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008

[12][S.J.K. Jagadeesh Kumar, "Performance comparison of QOS stability methods in WiMAX Networks" International Journal of Computer Science and Information Technologies, Vol. 1 (3), 2010, 169-174.

[13] White Paper, "TCP and Queue Management"