Volume 2 No. 1, January 2012
International Journal of Information and Communication Technology Research

http://www.esjournals.org

Favoring Short Flows Without Punishing Long Flows for the Internet Traffic Performance Improvement

[1]Y.Suresh, [2]S.Arumugam, [3]M.A.Bhagyaveni
[1]Department of Information Technology,
Sona College of Technology, Salem,TamilNadu,India
[2]Nandha Engineering College, Erode, TamilNadu,India
[3]Department of Electronics and Communication Engineering.
Anna University , Chennai. TamilNadu,India

ABSTRACT

An explosive growth in business applications using the Internet have resulted in a strong demand for some notion of reliability or quality of service. During periods of congestion or failure, the quality of service of all flows is degraded. As a result, a strong need for service differentiation in the flows to provide service guarantees. Today the internet carries different types of traffic with the increased use of peer to peer, HTTP, FTP applications. Web traffic comprises short flows which requires low response time and long flows seen in the internet originate form peer to peer file sharing applications TCP is the dominating protocol that carries majority of the total internet traffic. Recent internet traffic measurement shows most of the TCP flows are short lived. The performance improvement in the internet traffic can be achieved by the advantages of scheduling algorithms to favor short TCP flows first. However long TCP flows competing against short TCP flows starve at some point. Hence, the proposed Dynamic Queue Scheduling (DQS) favors short TCP flows without penalizing the performance of long flows using Dynamic packet scheduling ratio. We observed that the mean transmission time of flows and packet loss significantly decreases in comparison with FIFO and RuN2C.

Keywords: *Scheduling, Internet traffic, DQS, TCP flows*

1. INTRODUCTION

The studies [1],[2] shows that TCP conveys about 80-90 % of traffic over the internet Short flows are mainly generated by the delay sensitive applications such as Web, Telnet, VoIP [3].The long flows are generated in the internet originate form peer to peer applications[15]. Internet traffic exhibits that most of the TCP flows are short, while more than 50% of the bytes are carried by less than 5% of largest flows.[4].A flow is defined as a group of packets with a common set of attributes such as source address, destination address, source port, destination port.

Offering service guarantees to existing and emerging applications in the Internet has been a big challenge to Internet designers. One of the most important mechanisms to provide service guarantees is scheduling. Scheduling determines the order in which the packets from different flows are served. *Packet* scheduling in routers has been an active area of research in the last two decades, and most of the attention has focused on *Processor sharing* (PS) type of scheduling algorithms.

From queuing theory point of you it has been shown that choosing an appropriate scheduling algorithm significantly improves the performance of the system. The existing internet uses TCP as a Transport Control protocol and FIFO scheduling in routers The studies [5-9] shows that short flows should be given highest priority over the long flows. The issues in the design of a scheduling algorithm are a) Classification of short flows and long flows b) Favoring short flows without penalizing the performance of long flows.

Aiming these issues we propose a Dynamic Queue Scheduling algorithm (DQS) which uses a) Dynamic packet selection ratio, b)Fair treatment of short and long flows. TCP flows are classified a short and long and they are put in the two queues. Using Dynamic Packet Scheduling Ratio the packet in the two queues are served. This scheduling approach shows the fair treatment of short and long flows.

2. RELATED WORK

Two queue threshold based approaches has been proposed [5],[9] that gives highest service priority to the short flow. TCP flows are differentiated as short and long flows using a threshold value and short flows are en queued in one queue and remaining long flows are en queued in the second queue. Service priority is given to the first queue in First in First Out (FIFO) discipline and the second queue are only served if the first queue is empty. This approach reduces the mean transfer time however leads to starvation of long flows.

In RuN2C[6], using TCP sequence number the packets are put in first queue or second queue. Packets from the second queue were not served unless the first queue was empty. Limitation of this protocol is TCP sequence number should start from a set of possible initial numbers and would lead to security problems such as IP address spoofing and session hijacking .

In LAS[7], the next packet to be served is one belonging to the flow that has received the least amount of service. By this definition, LAS will serve packets from a

newly arriving flow until that flow has received an amount of service equal to the amount of least service received by flow in the system before its arrival. LAS is a preemptive scheduling policy that favors short jobs without prior knowledge of the job sizes. The long lived TCP flows competing against short TCP flows shows starvation in LAS. LAS may be extremely unfair for long flows starting during the service time of other long flows with almost the same flow size, because these long flows would be finished almost at the same time in spite of their different starting time. LAS reduces the loss rate for the short flows and approximately doubles the loss rate of long flows as compared with the loss rate under FIFO.

Another protocol Context Aware transport/Network Internet protocol (CATNIP) [10] requires application layer information, the web document size to provide explicit context information to the TCP and IP protocol. While this approach violates the traditional layered Internet protocol architecture, it enables informed decision-making, both at network endpoints and at network routers, regarding flow control, congestion control, and packet discard decisions.

The Queue State Packet Scheduling (QSPS)[11] uses a constant Packet Scheduling Ratio(PSR)to schedule sort and long flows . Packets are differentiated as short and long flows and using a constant PSR value service assignment is given to the packet in the two queues. The long flows in the second queue would not be starved since the queue does not have strict priority over the second queue. This constant Packet Scheduling Ratio cannot be applied for all network varying conditions.

3. ARCHITECTURE AND ALGORITHM DESIGN

Proposed scheduling algorithm DQS is suitable across varying traffic flows. The algorithm uses Dynamic Packet Scheduling Ratio *Q(r)* for the efficient scheduling. Here our classification of short and long flow is as follows: Short flows are those with the flow size less than the threshold th and otherwise long flows. Flow size is the total number of packets or bytes of the flow *i*. We divide queue into two groups: SFQ and LFQ . If a flow *i* and its bytes to be scheduled is less than the threshold *th* then the flow *i* is inserted in SFQ and if a flow *i* and its bytes to be scheduled is not less than the threshold *th* then the flow *i* is inserted in LFQ. On the arrival of a packet if the buffer if full, a selected packet will be dropped using buffer stealing [12],[13].Unlike Short Flow Highest Priority Scheduling algorithms ,we use Dynamic Packet Scheduling Ratio *Q(r)* to schedule the packets in the two queues. *Q(r)* decides the number of packets to be scheduled in each queue.

DQS- ARCHITECTURE

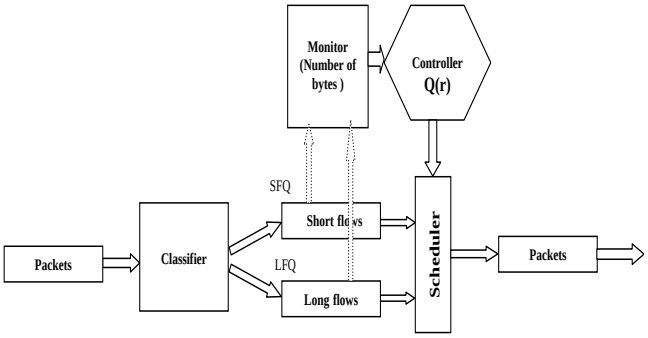

Figure:1 – DQS- Architecture

The Architecture components are explained as follows:

Classifier*:* On arrival of a packet *p* from flow *i* the classifier uses the threshold value *th* to each packet and dispatches it to the proper queue

Monitor**:** The Monitor statistics number of packets or bytes of all the flows in each queue and reports to the Controller periodically.

Controller**:** Controller is the soul of DQS. It calculates dynamic scheduling ratio *Q(r)* for the each round based on the bytes in two queue and commands scheduler to schedule the number of packets in each queue. It consists of a State Variable Counter *DCi*. The counter is initialized with the total bytes in SFQ. When the scheduler schedules the packets from SFQ, the *Q(r)* value is decremented from *DCi.* The controller makes decision according to the information that other components report.

Scheduler**:** The scheduler decides the service order of packets in two queues according to the Controller's command.

DQS –ALGORITHM

Begin:

Differentiate TCP flows as short flows and long flows using threshold *th* and insert in two queues SFQ and LFQ respectively.

S:

- Calculate bytes in SFQ = $\sum_{i=1}^{n} B_{SFQ}(i)$ and bytes in LFQ = $\sum_{i=1}^{n} B_{LFQ}(i)$
- Initialize a State Variable Counter $DC_i(r)$ =$\sum_{i=1}^{n} B_{SFQ}(i)$

- Calculate Dynamic Packet Scheduling Ratio *Q(r)*

$$Q(r) = \frac{\sum_{i=1}^{n} B_{SFQ}(i) + \sum_{i=1}^{n} B_{LFQ}(i)}{\sum_{i=1}^{n} B_{LFQ}(i)}$$

D:

- **If** $\sum_{i=1}^{n} B_{LFQ}(i)$ =0 **then** bytes scheduled in the queue LFQ = *Q(r)*
  else bytes scheduled in the queue SFQ = *Q(r)* and bytes scheduled in the queue LFQ =1
- erform$DCi(r) = DC_i(r) - Q(r)$
- he main observation is$DC_i(r-1) - Q(r) = DC_i(r)$.
- **If** $DC_i(r) > Q(r)$ **then** return to **D: else** return to **S:** for the calculation of Q(r) for the next round.

End

A State Variable Counter $DC_i(r)$ is initialized with a value of total number of bytes in the short flow queue SFQ .The bytes in the SFQ are scheduled using the calculated value of $Q(r)$. If $Q(r)$ value of bytes are scheduled in SFQ then one byte is scheduled in LFQ. This approach is continued until total number of bytes in the queue SFQ becomes zero and then LFQ is completely scheduled. The condition $DC_i(r) < Q(r)$ is checked each time when the packets are served in SFQ. If the condition is not satisfied then the new value of $Q(r)$ will be calculated for the next round.

This dynamic changing behavior of $Q(r)$ is the main difference between our approach and other threshold approach. The $Q(r)$ always changes according to the total number of bytes in the two queues. This adaptive ratio $Q(r)$ significantly improves the performance than the constant packet scheduling ratio used in QSPS [11] algorithm. The results shows that short TCP flows are treated without penalizing the performance of long TCP flows.

4. SIMULATION

The practical networks normally will have many bottleneck links interconnecting the router. Hence, the proposed Dynamic Queue Scheduling (DQS) is tested for their performance in the network topology with multiple bottleneck links. This model is shown in figure 2.In this configuration TCP sources traversing three bottleneck links and terminating at R3. The routers also shares the cross traffic .The bottleneck link capacities are 50Mbps, 30ms and other sources are connected with 10Mbps, 10ms. Here we refer to the packets that belonging to one TCP connection as a flow.

P

T

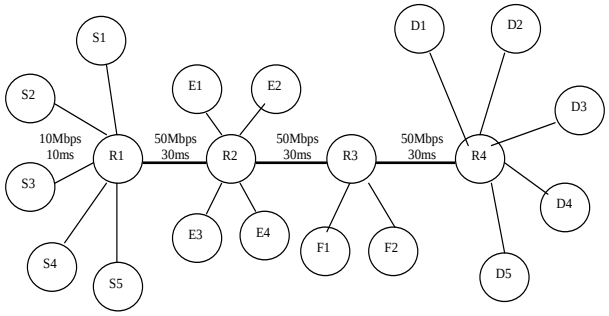

Figure:2 – Network Topology with Multiple Bottleneck Links.

For studying the performance of Dynamic Queue Scheduling (DQS), we test the following relations with the other protocols like First in First out (FIFO) and RuN2C.

4.1 Number of flows Vs Mean Transmission Time

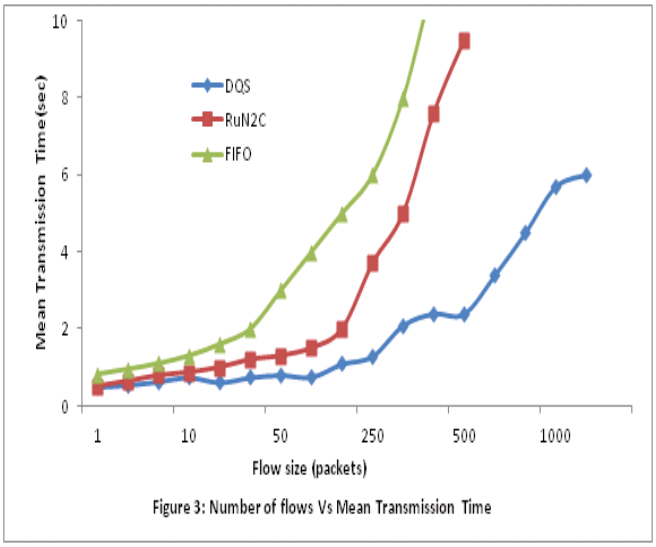

Figure 3: Number of flows Vs Mean Transmission Time

Figure 3 depicts DQS reduces the mean transmission time of flows by treating long flows fairly. The transmission time of a flow is time interval starting when a first packet leaves a server and ending when a last packet of flow is reduced by the corresponding client. It is evident from the figure that DQS approach reduces the mean transmission time compared to the simple FIFO and short flow highest priority scheduling. The transmission time of DQS and RuN2C are almost same up to the threshold DQS approach improves the performance for larger flows.

4.2 Short TCP flows Vs Mean Transmission Time

Figure 4: shows the mean transmission time of short flows. It indicates that the mean transmission time of

flows with flow size less than the threshold under DQS is almost the same as that under SFHPS . But DQS shows better performance for short flows larger than the threshold From Figure 4. we observe both DQS and RuN2C significantly reduce the mean transmission time of short flows compared with FIFO.

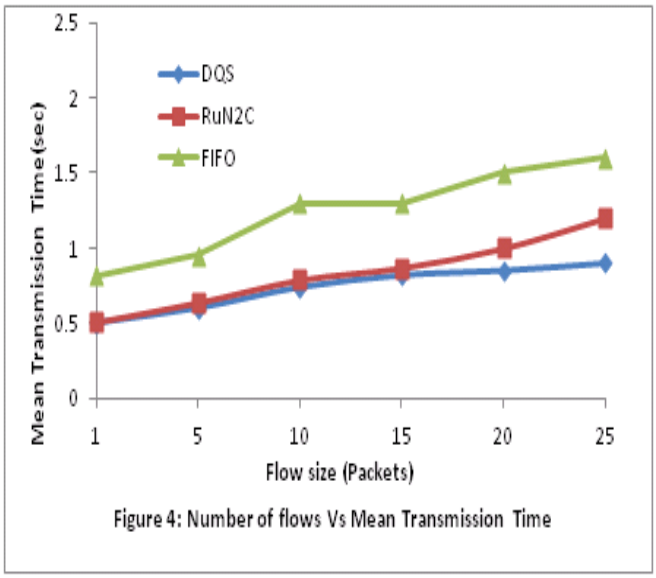

Figure 4: Number of flows Vs Mean Transmission Time

4.3 Comparison of Constant *Q* and Dynamic *Q(r)*

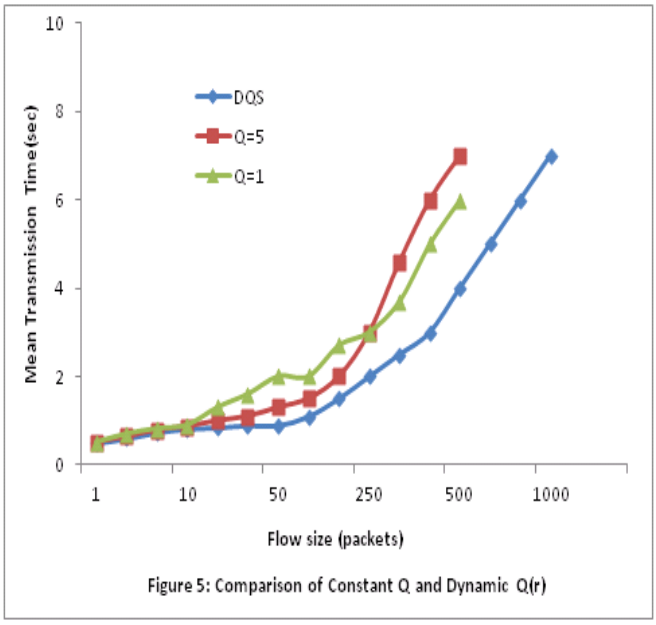

Figure 5: Comparison of Constant Q and Dynamic Q(r)

Figure 5.Depicts comparison of Constant *Q* and Dynamic *Q(r)* in terms of the mean transmission time of number of flows using various constant Packet Scheduling Ratio values. When *Q* =1 increases the mean transmission time of large flows because the packets in two queues are served with the equal priority. When Q value is increased to 5 the mean transmission time reduces up to the threshold value *th* and it is increases during large TCP flows. But the Dynamic *Q(r)* decreases the mean transmission time during large TCP flows. The mean transmission time is almost same up to the threshold when *Q*=5 and in DQS. This shows DQS algorithm treats short flows fairly without penalizing the performance of long TCP flows.

4.4 Number of Packets Dropped Vs Flow size

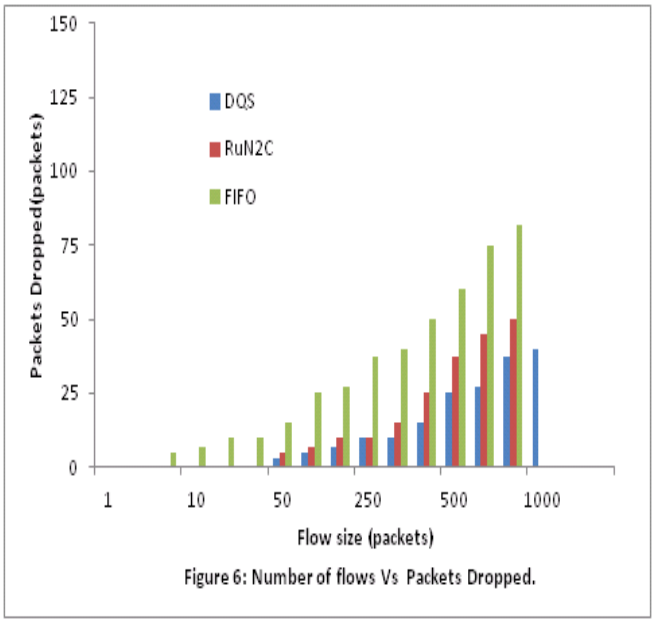

Figure 6: Number of flows Vs Packets Dropped.

Figure 6 shows the number of packets dropped for the various flow size. It is evident from the figure that short flows of size less than 40 packets not experiences packet loss in DQS, whereas for the similar size of flows FIFO experience packet loss. The packet loss is less in DQS compared to RuN2C for the large flows because of adaptive nature of the scheduling ratio used in DQS. The FIFO and Ru2Nc approach schedules the packets not considering the large flows but in DQS packets are scheduled from two queues. Hence large flows are also getting service in addition to the short flows. This reduces the packet loss in large flows.

5. CONCLUSION

Scheduling has been known for several years and attention has been given to use scheduling for the packet switched networks. In this paper we presented a Dynamic packet scheduling approach namely DQS to improve performance of short flows without penalizing long flows much. Unlike other scheduling approaches, the TCP flows are scheduled with a Dynamic Packet Scheduling Ratio *Q(r)* which decreases the mean transmission time and packet loss of flows compared with the other protocols like FIFO and RcN2C scheduling. This algorithm can be deployed in edge router without complexity.

REFERENCES

[1] Claffy, K., M.G., Thompson, K.: The nature of the beast: Recent traffic measurements from an internet backbone. In: Proceedings of INET '98, July 1998.

[2] Nandy, B., et al.: Intelligent traffic conditioners for assured forwarding based differentiated services networks. In: Proc. IFIP High Performance Networking, HPN 2000, Paris (2000)

[3] S.Floyd, "A Report on Recent Developments in TCP Congestion Control", *IEEE Communications*, Vol.39,No.4, pp.84-90, April 2001

[4] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modelling", *IEEE/ACM Transactions on Networking*, 3:226–244, June 1995.

[5] X. Chen and J. Heidemann, "Preferential treatment for short flows to reduce web latency," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 41, no. 6:779-794,2003

[6] K. Avrachenkov, U. Ayesta, P. Brown, and E. Nyberg, "Differentiation between short and long TCP flows: predictability of the response time," in *Proc. IEEE INFOCOM*, March 2004.

[7] I. A. Rai, E. W. Biersack, and G. Urvoy-Keller, "Size-based scheduling to improve the Performance of short TCP flows," *IEEE Network*, January/February 2005

[8] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 835-846, December 1997.

[9] L.Guo and I.Matta, "The War between Mice and Elephants", *In Proc. 9th IEEE International Conference on Network Protocols (ICNP)*, Riverside, CA,2001

[10] *C. Williamson* **and** *Q. Wu*, **"**A Case for Context-. Aware TCP/IP", *ACM Performance Evaluation* Re- view, Vol. 29, No. 4, pp. 11-23, March 2002.

[11] S.Arumugam, M.Chandrasekaran,Y.Suresh, J.Senthilkumar and V.Mohanraj "A QSPS –PSR Approach on Differentiated TCP Flows for Internet Traffic" *International Journal of Information Technology & Knowledge Management* Vol-III, Issue-I of June 2010

[12] M. Shreedhar and G. Varghese, "Efficient fair queuing using Deficit Round Robin," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 375-385, June 1996

[13] P. E. McKenney, "Stochastic fairness queuing," in *Proc. IEEE INFOCOM*, 1990

[14] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger,"Dynamics of IP traffic: A study of the the role of variabuility and the impact of control", In *Proc. of the ACM SIGCOMM*, August 1999.

[15] I. A. Rai, E. W. Biersack, and G. Urvoy-Keller, "Size-based scheduling to improve the performance of short TCP flows," *IEEE Network*, January/February 2005.