Available online at www.scholarsresearchlibrary.com

Scholars Research Library

European Journal of Applied Engineering and Scientific Research, 2012, 1 (4):91-98
(http://scholarsresearchlibrary.com/archive.html)

ISSN: 2278 – 0041

Verifying Queue Length Scheme in Wired Communication for Congestion Control

Manoj Kumar Prajapati; Divakar Singh

Dept. of CSE,BUIT,BU, Barkatullah University,Bhopal Bhopal (M.P), India

---

ABSTRACT

With the continuous increasing demand of internet applications. Network`s are expressing a serious congestion problem. In very large networks with heavy load traffic, Internet routers play an important role during congestion. All the internet routers have some buffers at input and output port which holds the packets in the time of congestion. Many queue management algorithms have been proposed but they mainly focus on fixed queue limit. Recognizing the fact that drop tail algorithm has fix maximum queue limit ,we direct our attention to variable Length queue limit for Combined Input and Output queued (CIOQ) switches. We propose a simple modification to the drop tail algorithm in which a generic queue management controls methodology in TCP/IP networks, that case we dynamic change queue length in our wired network. The performance of the proposed controlled system is evaluated via NS-2 simulator.

Keyword: congestion control, variable queuing.

INTRODUCTION

The continuous increasing demand for data transfer through the internet is exceeding the available network resources. This is mainly due to different applications that run concurrently and require high transfer data rates.TCP protocol which is the most important and popular protocol for data exchange through the internet, controls network congestion via measured packet loss rate. Congestion control in current Internet still is a critical issue. The number of Internet users is rapidly increasing and therefore the amount of data to be carried also increases. The main problem of Congestion occurred when arrival rate to a router is greater than its departure rate. Each router in the network uses queue management (QM) and scheduling as two classes of algorithms that are related to congestion control. A queue management system is used to control queues. Queues of people form in various situations and locations in a queue area. The process of queue formation and propagation is defined as queuing theory. Queues exist in two main forms. The QM algorithms try to control the length of packet queues by dropping packets when appropriate. Scheduling algorithms on the other hand, determine which packet to drop next and which is to send and also used to manage the allocation of bandwidth among flows. Buffer management schemes at routers decide when to drop a packet and which packet to drop. The simplest queue management scheme is drop-tail, where each packet is treated identically. With drop tail, 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. Once a queue has been filled, the router begins discarding all additional datagram's, thus dropping the tail of the sequence of datagram. The loss of datagram causes the TCP sender to enter slow start. It's worth nothing that most of the Internet today still runs drop-tail gateways. The performance of drop-tail for TCP traffic under high degrees of statistical multiplexing isn't particularly well-understood.RED [1] is an active queue management scheme, which explicitly tries to monitor and contain the average queue size to be small- enough for delays to be small, but large enough for bottleneck link utilization to be high when busty traffic causes a temporary full in offered load (e.g., when a set of TCP sources cut down their sending speeds).In RED scheme, packets are dropped even before the buffer is full, in order to notify sources if congestion is building up. The sources can then reduce their window sizes (or rates), thereby preventing further packet loss. However, RED has several shortcomings, including a high degree of sensitivity towards its operating parameters, unfairness to flows with different round-trip times, and the problem of global synchronization. Several studies have been performed to address the above shortcomings [9]. The idea of adaptively varying RED parameters has been proposed. The objective is to reduce the oscillations in the queue length. A-RED attempts to tune the RED parameters for a robust behavior but fails to do so in various dynamic cases as A-RED retains RED's basic linear structure. Random early detection (RED), also known as random early discard or random early drop is an active queue management algorithm. It is also a congestion avoidance algorithm

In this paper, we propose a variable length virtual output queue based congestion control mechanism. In this approach the maximum Queue Limit of Drop tail algorithm at each virtual output Queues is varying according to the no. of packets in each virtual output queues.

II.SWITCHING SURVEY

Typical queuing schemes for managing buffers inside packet switches include output queuing (OQ), input Queuing (IQ), virtual Output queuing (VOQ), and combined input and output queuing (CIOQ)[9].

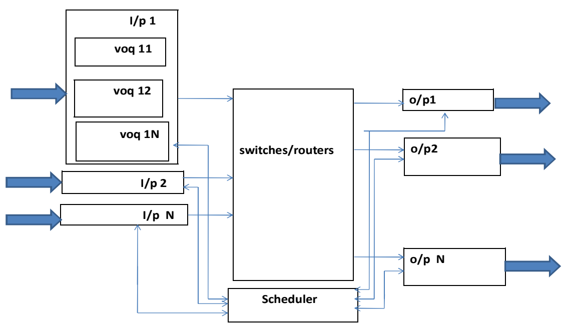

Fig.1.The CIOQ switch architecture

The output queuing scheme only uses buffer at each output module with no buffer at any input module. A packet switch using output queuing scheme is called an OQ switch. In an OQ switch, the switching fabric forwards any incoming packet to an output buffer immediately during its arrival time slot. However, a well-known limitation of output queuing is that in a switch with N ports, the switch must have an internal fabric speed that is N times the speed (capacity) of a link: N packets destined to some output may arrive at same time-step at different inputs. The switch fabric must be able to simultaneously transfer the N packets to that output port (i.e., at N time the speed of the switch links). This limits the applicability of output queuing in current switches where scalability, in terms of link speed and the number of ports, is primary design objective. The Input queuing scheme uses buffers only at each input module and no buffers at any output module. A packet switch using an input queuing scheme is called an IQ switch. The switching fabric of an IQ switch operates at the same rate as input ports and will not increase with the switch size. However, because there are no buffer at the output module, if there are multiple packets heading to the same output module, in each time slot only one of them can be forwarded through the switching fabric and all other are blocked at their arrival input modules. This causes the head of line (HOL) blocking problem [4, 8] in IQ switches if input buffer are FIFO queues.

Virtual output queuing scheme provides a solution to HOL blocking in IQ switches.VOQ scheme uses separated logical Queues at input port modules, one for each output port. A packet switch using virtual output queuing scheme is called a VOQ switch.

The combined input and output queuing scheme uses buffers at both input and output modules, and a switch that employs this queuing scheme is called a CIOQ switch. Fig. 1 illustrates the conceptual model of an N×N CIOQ switch with finite buffers. Every input port maintains a queue for each output port.

Fully wired communication networks:

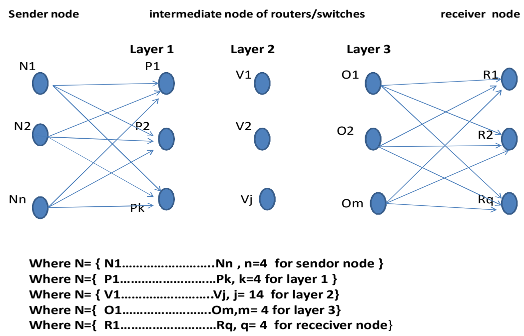

Fig. 2 Fully wired communication networks

III .PROPOSED SCHEME

We have proposed variable queuing scheme using VOQ method. In this method two scheme used in any intermediate forwarded node time, we use multiplexing and demultiplexing.

In case of multiplexing, multiplexing is the process where channels are combined for transmission over a common transmission path is called multiplexing. In case of demultiplexing, a device that performs the multiplexing is called a multiplexer (MUX), and a device that performs the reverse process is called a demultiplexer (DMUX). In this scheme we are using some formula for calculating multiplexing.

$$P_k = \sum_{i=1}^{nt} (n_i)$$

If sender node n= 16 and intermediate node k=4 than we calculate total multiplexing through each intermediate layer 1 node is equation above.

$$t = \frac{n}{k}$$

Variable Length VOQ:

Our approach is based on the fact that we can vary the length of the VOQ during the processing time. As it is possible to vary the VOQ length, so we can modify the Drop tail algorithm in which the maximum Queue Limit of the buffer size is fixed. The amount of variation in the VOQ's (while the total buffer size of an input port fixed one) is calculated. For example here are three possible cases for 2×2 switch (see fig.3).

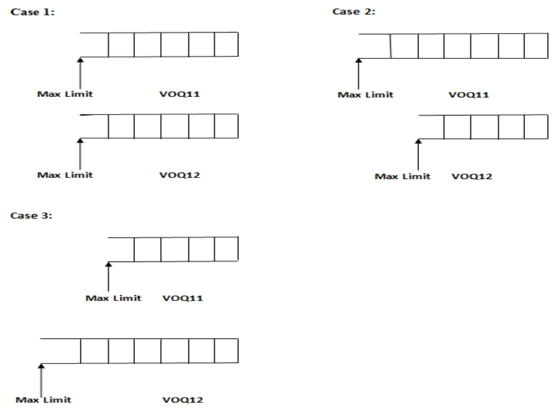

Fig. 3 In Case 1 the maximum Queue limit of both VOS's are equal. Case 2 shows that maximum Queue limit for VOQ11 is less than the VOQ12 and Case 3 shows that maximum Queue limit for VOQ11 is greater than the VOQ12.

According to above diagram some important linguistic rules for Input ports are obtained.
If voq11 is less and voq12 is also less then voq11_modified_queue_ length is equal to max queue limit.
if voq11 is avg and voq12 is less then voq11_modified _queue_ length is greater than max queue limit.
if voq11 is more and voq12 is less then voq11_modified _queue_ length is greater than max queue limit.
If voq 11 is less and voq 12 is avg the voq11 _modified_queue_length is lesser than max queue limit.
If voq11 is avg and voq 12 is avg then voq11 _modified_queue_length is equal to max queue limit.
If voq11 is more and voq 12 is avg then voq11 _modified_queue_length is greater than the max queue limit.
If voq 11 is less and voq 12 is more then voq11 _modified_queue_length is lesser than max queue limit.
If voq11 is avg and voq 12 is more then Voq11_modified_length is lesser than max queue limit

Simulation results:
In this Section, we compare the simulation results of our proposed scheme with the existing queue management schemes, drop tail and RED. All simulation is performed using NS-2 simulator. In all our simulation, we use the topology shown in figure 4(b).The buffer size of the input and output port is 120 and 100 packets respectively. Each input port carries multiplexed TCP Reno flows consisting of 1000 byte packets. The TCP flows are generated at separate source nodes then multiplexed together onto the backbone. Here we use 4 source nodes for each input port generating the same number of TCP flows varying from 150 to 500. The Queue monitoring interval is set to 0.0001 sec.

TCP sources

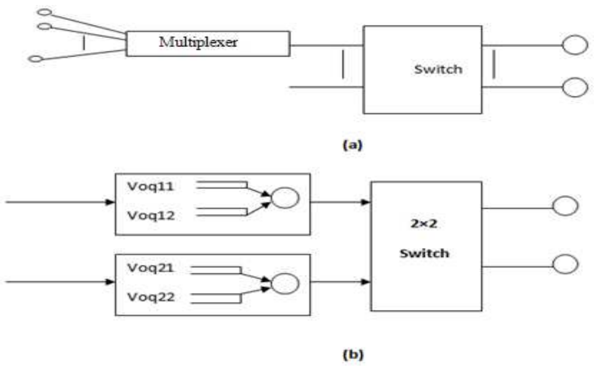

Fig 4 (a) General Topology of the simulated Network, (b) CIOQ with Virtual output Queuing Multiplexer

I. SIMULATION PARAMETERS AND PERFORMANCE EVALUATION

Table 1

Queue type	Drop tail, red, voq
No. Of sender node	4
Layer 1 intermediate node	16
Layer 2 intermediate node	14
Layer 3 intermediate node	4
No. Of receiver node	4
Queue length initial	50
Variable	±5

Input Queue length: Fig.5 displays the average input queue lengths when the modified algorithm is used in the QM unit for the input buffer, for the 2 × 2 switch. The corresponding simulations were performed for original RED and Drop tail, and the results are shown for comparison. For this simulation, 250 TCP sessions are started. Speedup is varied from 0.5 to 2.0. For RED, the minimum threshold is set to 17 packets and the maximum threshold is set to 49 packets. At output port in all cases we used the drop tail algorithm.

On the input side, as expected, the Drop tail queue has the longest average queue length, since it only starts dropping when the buffer overflows, whereas RED starts dropping packets before that. Also, the input queue length for suggested algorithm is roughly the same as drop tail algorithm.

Loss Rate: We also measured the loss rate of the 2 × 2 switch. Speedup is fixed and the load is varied from 150 to 500 TCP sessions. Loss rate is the ratio of the number of packets dropped and the number of packets sent. From Fig. 6, we observe that the performance of suggested approach is better than the drop tail and RED algorithm.

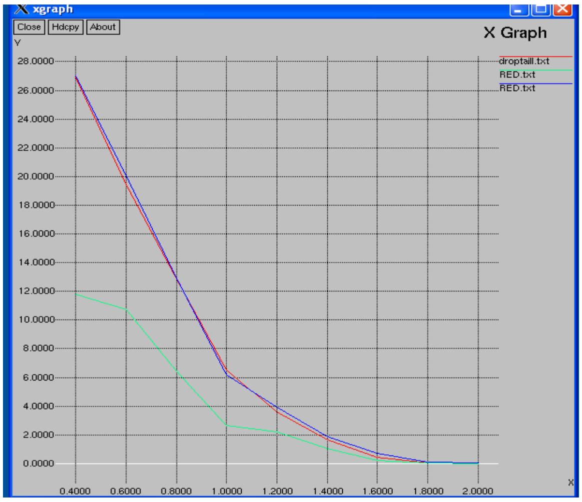

Fig 5. Average Input Queue Lengths v/s Speedup for 2×2 Switch (load 200 TCP Session)

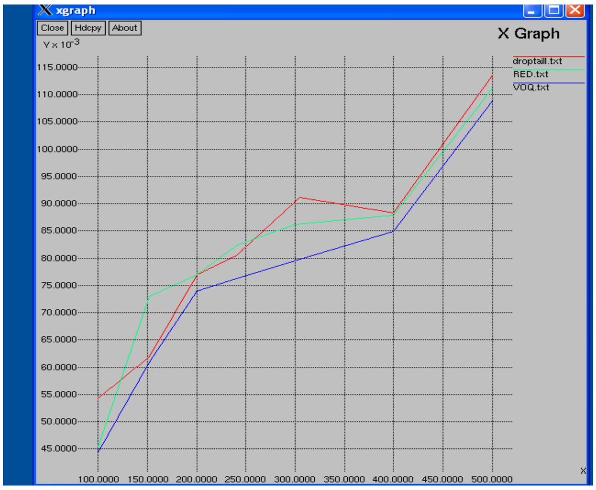

Fig6. Loss Rate of the 2×2 Switch at speedup1.1

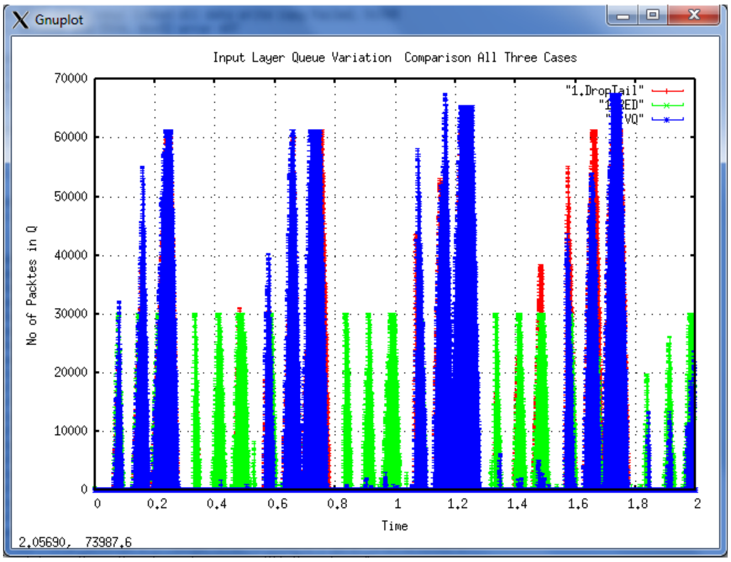

Fig7. Input layer queue variation comparison of all three cases

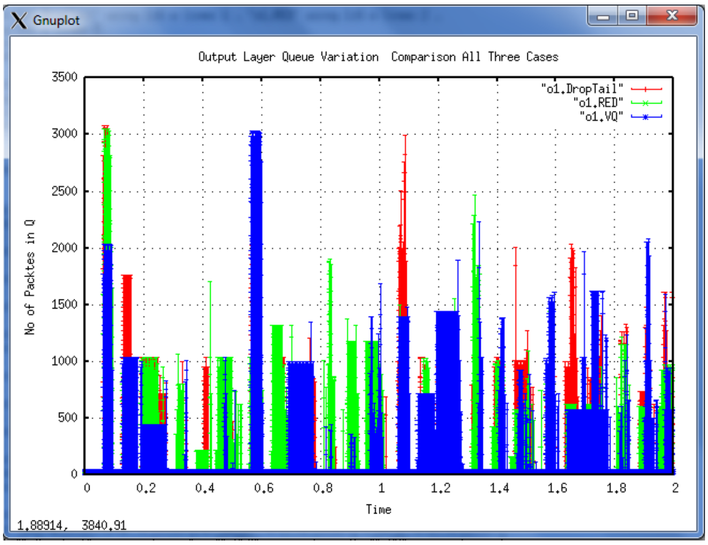

Fig8. Output layer queue variation comparison of all three cases

In Fig7. we investigate the Input layer queue variation comparison of all the three cases. Here blue coloured graph shows highest value for queue variation incase of varying queue.Green colored graph used for RED gives lowest queue variation. In Fig.8 we investigate the Output layer queue variation comparison of all the three cases. Output queue varies deliberately in case of varying queue shows highest variation than Droptail and RED. In Fig.9 TCP packet Drop analysis for all Q cases is shown.Maximum packet dropped in RED and lowest drop rate is in blue coloured graph representing varying queue.

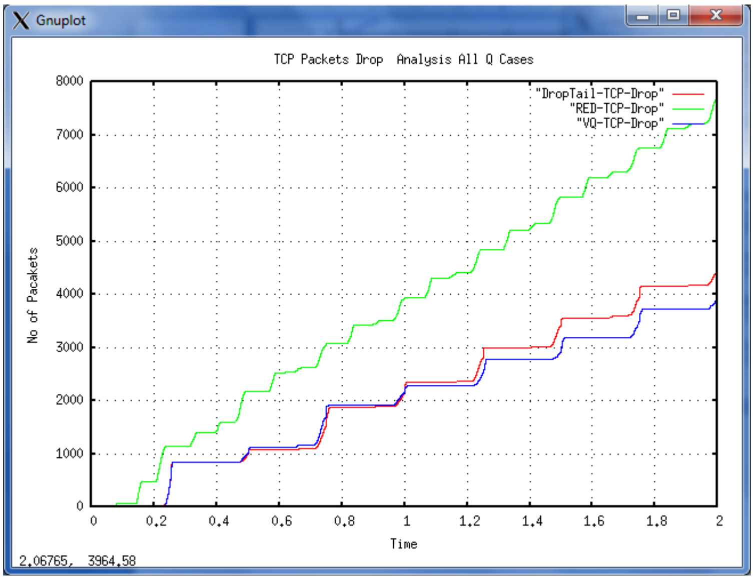

Fig9. TCP packet Drop analysis all Q cases

CONCLUSION

In this paper, we formulated an effective and efficient technique for queue management control, to solve the problem of congestion in TCP/IP networks. By extensive simulation, it is clear that small change in virtual output Queue length (varying maximum Queue limit For drop tail policy) while the buffer size of the input port is fix gives the good performance. The proposed scheme shows the improvements as compare to drop tail, and RED algorithm in terms of drop rate ,and buffer utilization in dynamic network environment (such as topologies and traffic condition).

REFERENCES

[1] Floyd S, Jacobson V, "Random early detection gateways for congestion avoidance" *IEEE/ACM Trans. on Networking*,1(4): 397–413,**1993**.

[2] C. Chrysostomou, A. Pitsillides, L. Rossides, A. Sekercioglu, "Fuzzy logic controlled RED: congestion control in TCP/IP differentiate services networks",Springer-Verlag 8(2):79-92,**2003**

[3] A. M. Al-Naamanyand H. Bourdoucen, "TCP Congestion Control Approach for Improving Network Services"**,** Springer Science+Business Media 13(1):1-6, **2005**.

[4] Nick McKeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches" *IEEE/ACM Tran on Networking*, 7(2):188-201, **1999**.

[5] L.A. Zadeh, "Outline of a new approach to the analysis of complex system and decision process", *IEEE Transactions on Systems, Man, and Cybernetics* 3(1):28-44, **1973**.

[6] Behrouz Safaiezadeh1, Amir Masoud Rahmani², Ebrahim Mahdipour3" A New Fuzzy Congestion Control Algorithm in Computer Networks" International Conference on Future Computer and Communication **2010**.

[7] Network Simulator, NS-2, 1989. <http://nsnam.isi.edu/nsnam/>.

[8] T.E. Anderson, S. S. Owickiet.al., "High Speed Switch Scheduling for Local Area Networks" ACM, **1992**

[9] Behrouz Safaiezadeh1, Amir Masoud Rahmani², Ebrahim Mahdipour3" A New Fuzzy Congestion Control Algorithm in Computer Networks" International Conference on Future Computer and Communication **2010**.