Active Queue Management in a QOS Enabled Network

P. Chenna Reddy[1], **Dr. P. Chandra Shekhar Reddy**[2]

ABSTRACT

IP provides only best efforts service but multimedia and real time applications require Quality of Service (QOS). Queue management algorithms which operate at the core of the network and that choose which packet has to be dropped, in case of buffer overflow or when the buffer size exceeds certain threshold make a significant contribution to QOS. DropTail is the traditional queue management algorithm, which drops packets when queue is full. This has two major problems, Lockout and Global synchronization that make it unsuitable for a QOS enabled network. RED is a proactive approach to congestion control. By providing advanced warning of incipient congestion, RED enables end nodes to respond to congestion before router buffer overflows and hence ensure improved performance.

This paper does the performance comparison of DropTail and RED and does an extensive study of RED. We examined the impact of both the Queue disciplines on TCP traffic. Performance difference is observed at higher levels of congestion. We measured the average queuing delay of DropTail and RED. RED by maintaining average queue size small tries to minimize the delay, which makes it suitable for delay sensitive applications. We then compared RED Byte and Packet modes of operation. The effect of RED's ability to mark packets instead of dropping them called Explicit Congestion Notification is also studied. The simulation is performed using NS2 and graphs are drawn using Xgraph.

KEYWORDS

Queuing, DropTail, RED, audio, video, ECN, and NS2.

1. INTRODUCTION

The Queue is a memory portion that is present in and around network switching elements and which can be used to temporarily store the packets before being forwarded. The process of temporarily storing the packets is called Buffering. The significance of buffering increased recently particularly with the transfer of data traffic over the Internet. The data traffic is bursty in nature and whenever there is a burst of data that cannot be transferred immediately buffering is required. The advantage of buffering is that it increases the network throughput, because logically the larger the buffer, the higher the amount of data carried by the network.

Buffering can be done at the input interface [4], output interface, and with in the switching fabric. Input interface is a switch component that receives packets from their intermediate nodes and selects the best output link in order to send the packet. The output interface is in charge of selecting the best packet in order to send it to the outgoing link. The switching fabric is a component that transfers packets from the input link to the output link. Mix of all the mechanisms can also be used but in general buffering is done at the output interface.

[1] Asst. Prof in CSE, JNTU College of Engg, Anantapur, Andra Pradesh. E-mail : pcreddy1@rediffmail.com

[2] Assoc. Prof and Head of ECE, JNTU College of Engg, Anantapur, Andra Pradesh.
E-mail : pcreddy21@rediffmail.com

Queue management is a mechanism that chooses which packet has to be dropped, in case of buffer overflow or when the buffer size exceeds certain threshold. Each queue in a network-switching node has a queue management mechanism associated with it. If a node has more than one queue there will be several instances of the queue management that operate independently on each queue.

Traditional queue management algorithm is DropTail in which packets are dropped from the end of the queue when there is no more space in the queue buffer. DropTail operates on each queue independently. There are several disadvantages of DropTail; Lock out and Global Synchronization being the prominent among them. With the demand for QOS, Internet needs an Active Queue Management algorithm, which takes into consideration dynamic traffic patterns and adjusts dynamically. Random Early Detection (RED) is one such prominent algorithm, which is recommended by IETF.

The benefits of AQM are several. First of all it allows reducing the number of packets dropped in the router, because keeping the average queue size small the router buffer can absorb bursts without discarding packets, or better, dropping fewer packets in the case of queue overflows. This is important not only because dropped packets represent a discard of resources, but also because TCP recovers easier from a single packet than a burst of packet drops. Second it provides lower delay on the link thanks to the reduced size of the buffer queue. This is very important particularly for applications that require low (maximum and average delay), examples being audio and video applications.

The characteristics of queue management algorithms are the following [1]:

- Fairness: Should treat all the flows based on the priority or weight assigned, otherwise treat all equally.
- Isolation: One flow should not affect other flows.
- Utilization: Should transfer more number of packets on to the output link and drop less number of packets.
- Accommodate bursts of data: The average queue size should be kept low and fluctuations in the actual queue size should be allowed, thus accommodating bursty traffic. Buffering in the network has to work in order to absorb data bursts and transmit them in a period of low link utilization. This means that the queue limits should reflect the size of burst we want the network to be able to absorb.
- Delay: Maintaining queue size small reduces the end-to-end delay i.e. both maximum and average delay.

The rest of the paper is organized as follows: Section II does the theoretical comparison of DropTail and RED and describes RED's byte and packet modes of operation and the methodology of marking packets called Explicit Congestion Notification. Section III presents the performance results. Section IV concludes the paper.

2. Theoretical Comparison

The DropTail is the simplest queue management algorithm, where packets are simply dropped when there is no more space in the buffer. RED (Random Early Detection) is an active queue management algorithm, which monitors the queue continuously and drops the packets early, before the problem becomes severe. Though RED drops the packets early, on the average RED drops less number of packets than DropTail.

The approach taken by DropTail i.e. waiting till the end has several limitations:

- Since packets are discarded at bursts (when the queue is full all the incoming packets are dropped) the utilization of the network fluctuates a lot. For example, when all the flows are of TCP type, TCP in response to dropped packets reduce the data rate resulting in a phenomenon called "global synchronization" where all the flows reduce the data rate resulting in decrease in utilization.

- Since congestion is detected only after a packet has been discarded and this occurs just when the queue router is full, DropTail allows queues to reach full status and to persist in this state for long.
- The data traffic is bursty in nature and the main purpose of queue is to absorb these bursts of data, but DropTail fails to absorb data bursts.
- DropTail mechanism may allow a monopolization of queue resources by a single or few flows, denying other connections the possibility to find place in the router buffer.

RED keeps traces of the past states of the queue. RED makes decision based on what is called as exponential moving weighted average function, which is able to differentiate a short term burst and long term congestion [2] [3]. Short-term bursts are quite common in the Internet traffic. In this case the algorithm should absorb short-term bursts and not drop packets. Algorithm must intervene when the number of packets in the queue is above the threshold, for a certain amount of time. This means that the network could become overloaded. For this it makes use of two parameters min_{th} (minimum threshold) and max_{th} (maximum threshold) and computes the moving average queue size. When average queue size reaches min_{th} it may drop some packets or make use of ECN (Explicit Congestion Notification) facility until the average queue size reaches max_{th}, after that it drops all the packets.

RED objective is to provide control on the average buffer in order to guarantee the following benefits:

- Avoid congestion by dropping packets early (early drops) or making use of ECN facility.
- Minimize packet loss and queuing delay.
- Avoid global synchronization of sources.
- Maintain high link utilization and maximize the global power (the ratio of throughput to delay)
- Allow bursty ness in traffic.

A.. RED byte mode Vs packet mode – To attempt to control incipient congestion, a RED router must somehow determine the amount of congestion currently occurring in order to apply a suitable amount of backpressure on the data senders. RED uses average queue length for the purposes of gauging the congestion state of a particular link. Queue length can be measured in two ways [8]: using the number of packets awaiting service in the queue, or the number of bytes sitting in the queue waiting to be forwarded. The choice of metric has implications on the traffic shaping applied by RED.

If a router stores packets in fixed buffers regardless of packet size a 50 byte packet and a 1500 byte packet take the same amount of internal router resources. On the other hand, if packets are stored in a single large memory buffer in the router and take only the amount of memory they need, then the 1500 byte packet takes 30 times more queue space than the smaller packet. Additionally, the delay through a router is dictated by the size of the packets in the queue (i.e., the number of bytes that must be serialized). There are tradeoffs to measuring the queue in terms of Bytes or Packets. Some of these tradeoffs involve specific router architecture (e.g., memory allocation issues), while others are more generic.

B. Explicit Congestion Notification (ECN) - The Active queue management algorithm detects congestion before the buffer queue overflows and signals to the end nodes the incipient congestion. As mechanism for congestion indication the router may drop packets, but a more efficient way to achieve this is marking packets. When the average queue size exceeds minimum threshold, instead of dropping packets, packets are marked. One significant point to remember is the reaction of the senders to dropping packets and also to marking packets is same.

Provision is made for ECN [6] [7] [9] in the IP header. In the TOS octet of IP header, the ECN field is defined through the bits 6 and 7, with bit 6 standing for ECN capable transport (ECT) and bit 7 for congestion experienced (CE) bit. ECT bit is used to indicate whether end points are ECN capable or not and CE code point is used to signal the presence of congestion. ECN field in IP is as shown in Table 1.

ECT	CE	Names for ECN bits
0	0	Not-ECT
0	1	ECT (1)
1	0	ECT (0)
1	1	CE

Table 1: ECN in IP

When the router is ready to discard packets to signal end-points of incipient congestion, at the reception of a new packet it has to check the ECT code point. If it is set (ECN-capable flows) it should set the CE code point and forward the packet, or on the contrary if it is not set (not ECN-capable flows), discard packet. When the end-nodes receive a packet ECN-capable with the CE bit set, the reaction will be same as detection of a packet drop. It is important that the router sets the CE code point only in the case it would have otherwise discarded the packet.

3. Simulation

The aim of our simulation is to compare DropTail with RED and do an extensive study of RED. RED is operated in both the Byte and Packet modes and RED's ability to mark packets instead of dropping (explicit congestion notification) is also experimented with. Our simulation is comprehensive because we measured all the performance parameters like Queuing delay, average queue size, not just number of packets delivered successfully or dropped.

A. Methodology - For simulation we used dumbell topology, TCP Vegas as the Transport agent with FTP as the traffic source except for the case where RED's ECN feature is experimented, for which we used TCP Tahoe with FTP as the traffic source. Identical loads are applied to achieve fairness in comparison. The simulation is performed for 100 sec.

For the case where RED's ECN feature is experimented we used both homogeneous traffic [10] (all the flows assigned same packet size) and heterogeneous traffic (different flows assigned different packet sizes).

Simulation is performed using NS2 [5] and the graphs are generated using X-graph. NS2 is a discrete event simulator developed by the University of California at Berkeley and the VINT project. NS2 supports two languages, system programming language C++ for detail implementation and scripting language TCL for configuring and experimenting with the different parameters quickly. NS2 has all the essential features like abstraction, visualization, emulation, and traffic & scenario generation. X-graph draws a graph on a display with data given either from data files or standard input. It can display up to 64 independent data sets using different colors and line styles for each set.

B. Metrics - The following is the list of quantitative metrics using which we did the performance analysis:

- Number of packets delivered successfully (throughput)
- Number of packets dropped
- average queuing delay
- average queue size

C. Simulation environment - The topology for our simulation is as shown in Fig. 1.

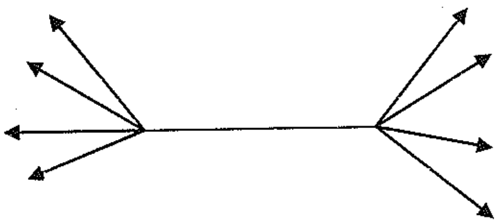

Figure 1: Dumbell Topology

The bandwidth of all the links including the bottleneck link is 10Mbs and the propagation delay is 1 msec. The queuing algorithm associated with all the links except bottleneck link is DropTail. The mean packets size is chosen as 500 as it is close to 576 bytes, which is the default size of IP packet.

D. Simulation results:

1) Throughput and Packet Drops comparison of DropTail and RED:

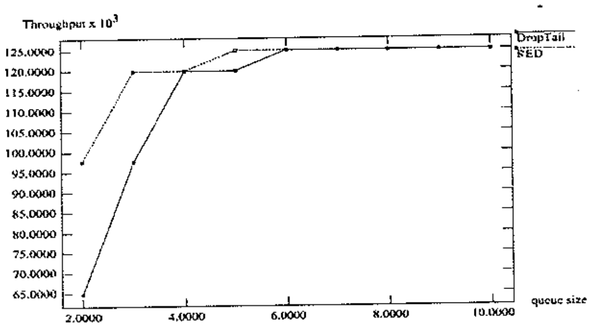

Figure 2: DropTail and RED Throughput Comparison

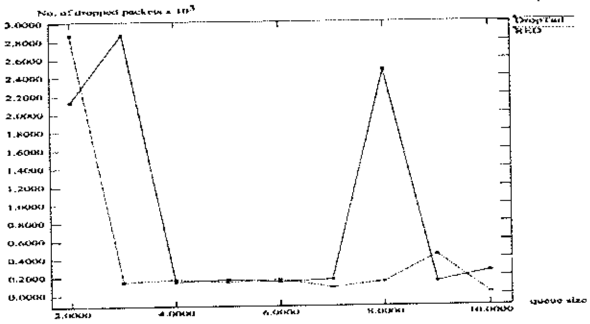

Figure 3: DropTail and RED Packet Drops Comparison

DropTail and RED performance is approximately same when the level of congestion is not significant. Difference in performance is observed when the level of congestion increases and RED throughput is better than that of DropTail as in Fig. 2. Though it appears in Fig. 3 that RED is dropping significant number of packets when the level of congestion is high, it does not result in decrease in throughput because it does not suffer from what is called as global synchronization. Since DropTail does not drop packets until the queue is full, it drops all the packets when the queue is full resulting in all the flows decreasing their data rate (global synchronization) and decrease in the overall throughput. For applications which require high throughput like File transfer, RED is better choice. Neither DropTail nor RED has the property of fairness.

2) Queuing Delay comparison of DropTail and RED: Delay is one of the Quality of Service requirements. The number of applications, which are sensitive to delay, is increasing day by day. Audio and Video applications are prominent among them. When audio is transferred over the Internet (VOIP) the average delay and maximum delay should be controlled, otherwise gaps of silence may be observed and it can be annoying for both the speakers. For video on demand where the video is played on line or for video conferencing, if delay fluctuates then flickering will be observed.

Traffic Source	DropTail Average Queue Delay	RED Average Queue Delay
1	0.007486	0.005011
2	0.007489	0.0049979
3	0.007493	0.005001
4	0.007497	0.0052617

Table 2: DropTail and RED Average Queue Delay

The average delay of RED is significantly lower than that of DropTail as shown in Table 2, the reason being RED controls its queue size and it doesn't allow queue size to reach the maximum limit. DropTail doesn't control the queue size and allows the queue size to reach the maximum value. As the queue size increases average delay automatically increases.

3) RED byte and packet modes comparison:

When homogeneous traffic is applied there is no significant difference in byte and packet modes of operation except that byte mode drops more number of packets as in Table 3.

Packet size	Throughput		No. of dropped packets	
	Byte mode	Packet mode	Byte mode	Packet mode
296	21121	21121	38	11
500	12519	12519	528	528
3000	2092	2093	188	48
13000	483	483	30	8

Table 3: RED with homogeneous Traffic

Packet sizes	Throughput		No of Dropped packets	
296 500 3000 13000	Byte mode	Packet mode	Byte mode	Packet mode
	6058032	6257960	488848	315328

Table 4: RED with heterogeneous traffic

There is difference (2%) in performance when heterogeneous traffic is applied and packet mode performs better as in Table 4. Byte mode drops more number of packets because as packet size increases, the probability of it being dropped increases. For example if the packet size is 1500 and the mean packet size is 500 then probability of it being dropped is three times the case when the packet size is 500.

4) RED performance with and with out ECN:

Explicit Congestion notification results in decrease in the number of dropped packets in both homogeneous traffic and heterogeneous traffic cases.

Packet size	Throughput		No. of dropped packets	
	With ECN	With out ECN	With ECN	With out ECN
296	18473	18479	65	804
500	11518	11511	74	565
3000	2047	2054	49	144
13000	485	485	18	28

Table 5: RED with homogeneous traffic

Packet Size	Throughput		No. of dropped packets	
296 500 3000 13000	With ECN	Without ECN	With ECN	With out ECN
	6114968	6219192	220868	524380

Table 6: RED using heterogeneous traffic

But there is no difference in throughput, in homogeneous case, because marking has same effect as that of dropping packets as shown in Table 5. The sources react identically to marking packets or dropping them. There is minor difference in throughput in heterogeneous case as in Table 6.

5) RED Average queue size computation:

RED gateway calculates the average queue size that determines the degree of burstiness allowed in the router queue. One of the main characteristics of RED is its average queue size is less as in Fig. 4, which allows it to absorb bursts.

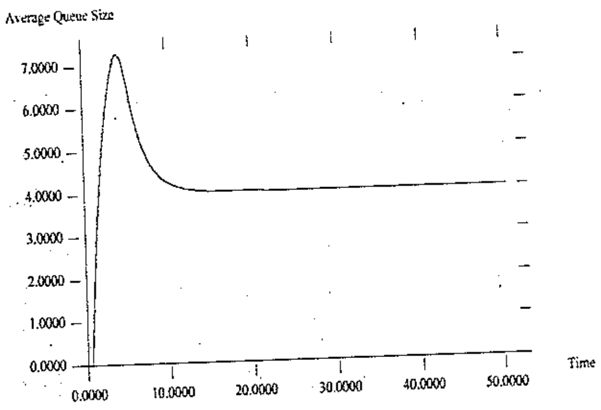

Figure 4: RED Average Queue Size Vs Time.

By keeping average queue size low it keeps room for burst of data to be absorbed. Significantly data traffic is bursty in nature.

4. Conclusion

Over the past few years, variety of queue management algorithms has been proposed. DropTail is the traditional queue management algorithm, and RED is an Active queue management algorithm which is being used in Internet. There are comparative studies on DropTail and RED but there is no comprehensive study, which evaluates all the features and operating modes of RED. Our study not only compares DropTail and RED, but also analyzed RED's performance in all modes of operation. We compared DropTail and RED at various levels of congestion and found that RED performs better at higher levels of congestion. At all the levels of congestion RED's average delay is better and makes it suitable for the current killer applications like Audio and Video transfer. We also evaluated RED's current experimental feature ECN. RED also maintains its average queue size with in limits allowing itself room for absorbing data bursts.

We are doing in parallel, the work related to the role played by end-to-end mechanisms and edge mechanisms in providing Quality of service. Our future work includes the study of two Quality of service architectures DiffServ and IntServ and to study the approaches to be followed for achieving QOS in Wireless Networks.

5. References

[1]. B. Braden, D. Clark, "Recommendations on Queue management and congestion avoidance in the Internet," RFC 2309, April 1998.

[2]. S. Floyd, V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM transactions on Networking, August 1993.

[3]. V. Jacobson, "Congestion Avoidance and Control," Proceedings of the SIGCOMM'88 Symposium, pp. 314-32, August 1988.

[4]. M. Karol, M. Hluchyi, S. Mogan, "Input versus output queuing on a space-division packet switch", IEEE Transactions on Communications, Vol. COM-35, pp. 1347-1356, Dec. 1987.

[5]. Network simulator v2 - NS2, http://www-nrg.ee.lbl.gov/ns.

[6]. K. Ramakrishnan, S. Floyd, D. Black, "The Addition of Explicit congestion notification (ECN) to IP," RFC 3168, September 2001.

[7]. J. H. Salim, U. Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", RFC 2884, July 2000.

[8]. Sally Floyd, "RED: Discussions of Byte and Packet Modes," http://www.icir.org/floyd/REDaveraging.txt, March 1997.

[9]. Sally Floyd, "TCP and Explicit Congestion Notification," Computer Communications Review, 24(5):10–23, October 1994.

[10]. Stefann DeCnodder, Omar Elloumi, and Kenny Pauwels, "Effect of Different Packet Sizes on RED Performance," in Proc of the Fifth IEEE Symposium on Computers and Communications (ISCC), 2000.

Authors' Biography :

Sri P.Chenna Reddy was born on 16-03-1973 at Kurnool. He did his B.Tech (CSE) from S.V. University College of Engineering, Tirupathi in the year 1996. He did his M.Tech from JNTU, Anantapur. Currently he is doing research on "QOS in Adhoc Networks".

He has a total of 9 years of experience in teaching. Currently he is working as Assistant Professor in JNTU, Anantapur. He has a total of 5 publications to his credit.

Dr. P.Chandra Sekhar Reddy was born on 15-08-1966 at Chinnamanchupalle, Kadapa. He did his B.Tech (ECE) from JNTU, Anantapur in the year 1983, M.Tech applied electronics from Baratiar University. He did his Ph.D. from JNTU, Anantapur in the year 2001 on "Routing in Adhoc Networks". He worked as trainee at ISRO for 10 months. He has a total of 15 years of experience. Currently he is Head of ECE Department at JNTU, Anantapur. He has published 4 papers in journals, 12 in conferences and he has also written 1 text book.