2010 2nd International Conference on Computer Technology and Development (ICCTD 2010)

The Study of ECN Application Effect at the Performance Improvement of RED

Alireza Gharegozi
Faculty Member of Computer Group
Islamic Azad University – Shahindezh Branch
Shahindezh, Iran
Gharegozi@yahoo.com

***Abstract*—Random Early Detection (RED) algorithm, a framework of active queue management (AQM), has been proposed in order to improve the performance of congestion routers due to the increased use of Internet. But, the basic problem of a lag between congestion's occurrence and its remedy remained. To overcome this problem of RED, Explicit Congestion Notification (ECN) proposed by the IETF that allows end-to-end notification of network congestion without dropping packets. This paper evaluates performance enhancement of RED with ECN Support using Standard RED as the evaluation baseline with respect to theirs abilities of improving performance of congested routers by keeping packet drop rate low. The performance is evaluated for FTP-like bulk-data TCP flows and UDP based traffic such as Constant Bit Rate(CBR) based on average throughput results of TCP sources by using NS2 Simulator. The simulation results show that use of ECN in light congested networks has not more benefits.**

Keywords-RED; TCP; ECN; congestion control; active queue management; average throughput;

I. INTRODUCTION

Congestion in network occurs due to exceed in aggregate demand as compared to the accessible capacity of the resources. Due to increase in Internet size and no. of users, clients are likely to experience longer delay, more packet loss and other performance degradation issues because of network congestion. Formally this problem was tackled by network service providers in terms of keeping utilization of the network low, which may regard as an infeasible solution. As the Internet is gradually dominated by the IP and packet switching, so to increase the network performance in terms of satisfactory level of service to clients is considered as challenging problem [1].

The Transmission Control Protocol is one of the core protocols of the Internet Protocol Suite. TCP is so central that the entire suite is often referred to as "TCP/IP." Due to network congestion, traffic load balancing, or other unpredictable network behavior, IP packets can be lost or delivered out of order. TCP detects these problems, requests retransmission of lost packets, rearranges out-of-order packets, and even helps minimize network congestion to reduce the occurrence of the other problems [2].

Drop Tail is the simplest and most widely used congestion control scheme in the current Internet routers. It works on first-in-first out (FIFO) based queue of limited size, which simply drops any incoming packets when the queue becomes full. Drop Tail and other schemes like Drop Front on full or Random Drop on full are unable to solve the problem of full queues [3].

To overcome these problems, one of the possible solutions is to detect congestion earlier and then accordingly to acknowledge the sources about congestion through congestion notification before queue gets overflow. The mechanisms who adopt this strategy are known as “Active Queue Management (AQM) Schemes” [4].

Random Early Detection(RED) [5] AQM algorithm has been proposed in order to alleviate the problems of simple drop-tail queue management. 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.

Explicit Congestion Notification(ECN) proposed by the IETF to reduce packet drop rate in the Internet. ECN is an extension to the Internet Protocol that allows end-to-end notification of network congestion without dropping packets. For networks with mechanism for the detection of incipient congestion, the use of ECN mechanism for the notification of congestion to the end nodes prevents unnecessary packet drops[6],[7].

This paper presents performance evaluation of RED with ECN Support using Standard RED as the evaluation baseline with respect to theirs abilities of improving performance of congested routers by keeping packet drop rate low. The performance is evaluated for FTP-like bulk-data TCP flows and UDP based traffic such as Constant Bit Rate(CBR) based on average throughput results of TCP sources by using NS2[8] Simulator. The evaluation is done in a number of Network scenarios.

II. RELATED WORK

A. TCP Congestion Control

The TCP congestion control consists of slow start, congestion avoidance, fast retransmit and fast recovery algorithms [2].

During slow start, a TCP increments congestion window (cwnd) by one segment for every ACK received. The slow start ends when cwnd exceeds the slow start threshold (ssthresh), and congestion avoidance takes over. During congestion avoidance, cwnd is increased linearly, i.e. by 1

978-1-4244-8845-2/10/ $ 26.00 © 2010 IEEE

segment per round-trip time (RTT). When a TCP sender detects segment loss using retransmission timer, the value of ssthresh is halved, and the cwnd is reset to 1 segment. The lost segment is retransmitted.

The TCP-sender uses fast retransmit to detect and repair, loss based on incoming duplicate ACKs. After receiving 3 duplicate ACKs, TCP retransmits the missisegment without waiting for retransmission timer to expire. After the fast retransmit, the fast recovery algorithm governs the transmission of new data until a non-duplicate ACK is received. After the receipt of third duplicate ACK, the value of ssthresh is halved and cwnd is set to ssthresh + 3 segemnts. The TCP algorithm is given in Fig. 1.

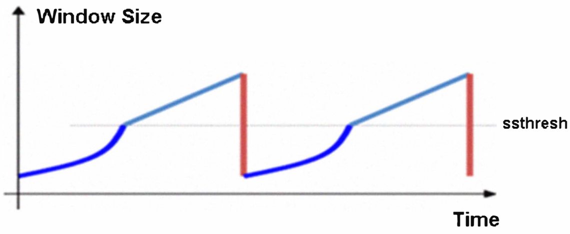
Over the years a lot of research concerning TCP has been carried out and many modifications and extensions such as Reno[9], New Reno[10], SACK[11] and Vegas[12] have been proposed.

B. RED : Random Early Detection

The Active Queue Management (AQM) algorithm is a solution to the problem of congestion control in the Internet routers. Random Early Detection (RED) is recommended as one of the mechanisms for AQM by IETF. RED was designed with the objectives to minimize packet loss and queuing delay and maintain high link utilization.

A router implementing RED AQM maintains a single queue to be shared by all flow that drops an arriving packet at random during periods of congestion. RED keeps the average queue size low, allows occasional packet bursts, and prevents global synchronization of source windows due to its randomness in marking or dropping packets at a congested node. However, it has been proven through simulations that an unresponsive bandwidth greedy connection gets a larger than fair share of the bandwidth at a bottleneck link when competing with responsive connections at a RED gateway. RED monitors the average queue size (avg) of a shared FIFO queue at the router output port. Based on comparison of the average queue size against a minimum (minth) and a maximum threshold (maxth) a packet dropping probability (pa) is computed. The RED algorithm is given in Fig. 2. A detailed explanation of the RED algorithm can be found in [5].

The RED based AQMs such as PD-RED[13], Auto RED[14], MRED[15], DS-RED [16] and FRED[17] tried to solve the various problems existing with RED.



- Slow Start
  - Start with W=1
  - For every ACK, W=W+1
- Congestion Avoidance
  - For every ACK,
  - W = W+1/W
- Congestion Control
  - ssthresh = W/2
  - W = 1

Figure 1. TCP Algorithm

```
For each packet arrival :
    Calculate the average queue size
  If q > 0
      avg = (1 - wq) avg + wq q
  else
      avg = (1- wq)m avg
    If minth <= avg < maxth
          Calculate packet dropping probability
          pb = maxp(avg - minth)/(maxth - minth)
          pa = pb /(1 - count pb)
          With probability pa Drop the arriving packet
    else if maxth <= avg
          Drop the arriving packet.
```

Figure 2. RED Algorithm

C. ECN: Explicit Congestion Notification

Explicit Congestion Notification is an extension to the Internet Protocol. ECN allows end-to-end notification of network congestion without dropping packets. Traditionally, TCP/IP networks signal congestion by dropping packets. When ECN is successfully negotiated, an ECN-aware router may set a bit in the IP header instead of dropping a packet in order to signal the beginning of congestion. The receiver of the packet echoes the congestion indication to the sender, which must react as though a packet drop were detected[6].

When ECN scheme is active, RED gateway is configured to mark rather than drop packets. On one side, the ECN scheme used in this report uses a single message as an indication of network congestion. On the other side the scheme tries to make sure TCP does not respond too frequently by reacting to congestion notification at most once per round trip time (this includes triple-acks).

Before any ECN-enabled data exchange can take place between two endpoints, they first have to successfully negotiate the use of ECN. ECN negotiation happens during the TCP connection setup phase. The ECN-related bits are (i) ECN-Capable (ECT) and (ii) Congestion Experienced (ECN/CE) bits in the IP header, and (iii) ECN-Echo bit in the TCP header. In moments of congestion the ECN-enabled router marks ECT-enabled packets by setting the ECN/CE bit in the IP header, as illustrated in the figure. When such packets reach the client, the client sets the ECN-Echo bit in the TCP header of the corresponding ACK packet thus signaling to the server that the incoming data packet has experienced congestion[7]. The ECN algorithm is given in Fig. 3.

III. SIMULATIONS

A. *General Simulation Configuration*

This paper categorizes traffic into the following two groups, according to the way the traffic handles congestion. Bulk data TCP flows and Cross data CBR flows. send from the left to the right over the network shown in Fig. 4.

The bulk traffic sources uses TCP (i.e., a flow with a large amount of data to send, such as FTP transfers) and send data continuously throughout the simulation. The cross traffic sources uses UDP, have exponentially distributed on and off periods with means of 2.5 seconds and send at 100 Kbps. The goal is to model bursty cross traffic that is not responsive to congestion, for example, bursty video sources.

The routers have 25 packet buffers. R1,input port of R2 and output port of R3 use FIFO DropTail queuing. FIFO DropTail routers queue packets first-in first-out and drop the last packet received when the queue is full.

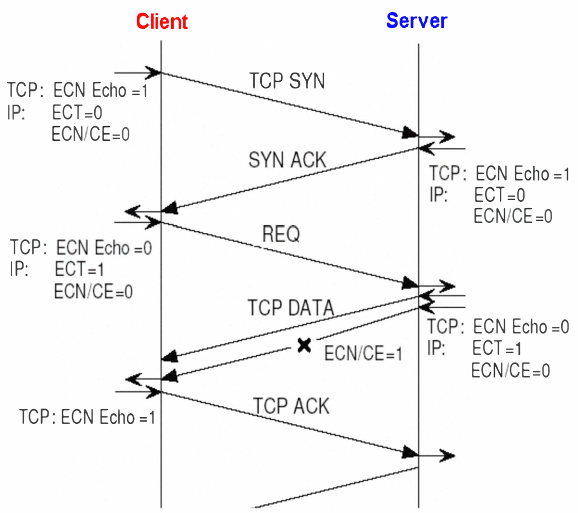


Figure 3. Negotiating ECN Capabilities

Output port of R2 and input port of R3 (the link between R2 and R3) use RED queuing. RED parameters are set as follows: minth= 6, maxth =18. All links from bulk traffic sources to R1 have a delay of 10ms and bandwidth of 1 Mbps, Other links from endpoints to routers have a delay of 10ms and bandwidth of 10 Mbps.

All endpoints use 1000 byte packets. Each simulation is repeated for different delays on the link from the R1 to the R2 between 25 and 500 ms, which models the bandwidth delay products ranging from a geographically small terrestrial network to a network with a (low-bandwidth) satellite link. This range was selected because it is common round trip times in the Internet.

All simulations were made using NS2, a simulator produced by the University of California Berkeley, the Lawrence Berkeley National Labs, and the Virtual Inter-Net Testbed (VINT) project[8]. All throughputs based on the number of useful packets received by the destination endpoint. Retransmitted packets are not considered useful.

B. *Experiments with Different Congestion Scenarios*

1) *Simulations without UDP Traffic(Light Congestion)*

The simulations in this section use the network simulation configuration in Fig. 4. The average throughput of the 10 TCP sources are plotted in Fig. 5 for various values of Delay. RED reduces the time period that a router is overloaded by causing the endpoints to infer that the router is congested before congestion at the router has become critical. But, the basic problem of a lag between congestion's occurrence and its remedy remains. Table 1 reports none improvement from using ECN for each delay value.

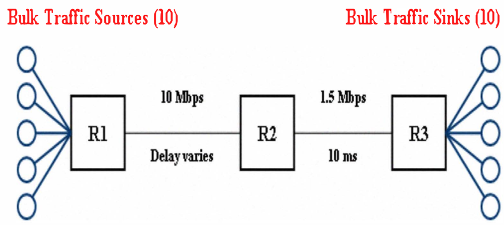


Figure 4. Standard Network Topology

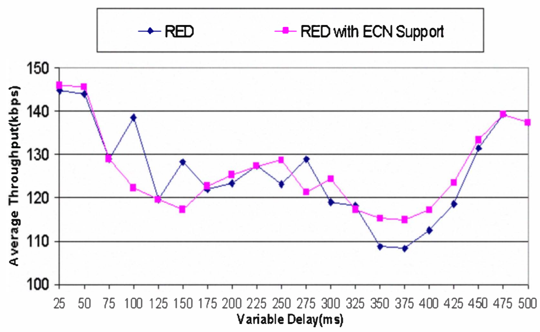


Figure 5. The average throughput of the 10 sources

TABLE I. THROUGHPUT STATISTICS FOR EXPERIMENT 1

Delay	Average Throughput	
	RED	***RED with ECN***
50	144.0208	145.6819
100	138.5811	122.3003
250	123.1177	128.7585
375	108.3196	114.9032

2) *Simulations with UDP Traffic(Heavy Congestion)*

The simulations in this section use the network simulation configuration in Fig. 6. Fig. 7 plots the average throughput seen by the bulk traffic sources against the variable delay. Each point is the average throughput of the 10 bulk transfer sources in the simulation. UDP flows are unresponsive because UDP does not react to network congestion. When congestion occurs, buffers could overflow and incoming packets will be discarded. Table 2 reports the beter performance using ECN in comparison to RED for each delay value.

IV. CONCLUSIONS AND FUTURE WORK

This paper presented performance evaluation of RED with ECN Support using Standard RED as the evaluation baseline with respect to theirs abilities of improving performance of congested routers by keeping packet drop rate low. The simulations in the experiment 1 showed that use of ECN in light congested networks has not more benefites on the average throughput of the 10 bulk transfer sources in comparison to the heavy congested network.

Although ECN leads to fewer packets drops, it does not necessarily lead to improved throughput for TCP transfers. On the other hand, in no case does ECN seem to lead to an actual degradation in TCP performance. In summary, the TCP/ECN sender has a competitive advantage over ECN-unaware senders because it reacts faster to incipient congestion and can thus avoid unsuccessful segment transmissions. Nevertheless, TCP/ECN has the same mechanisms as standard TCP for detecting improved network conditions, and, therefore, is unable to deliver guaranteed improved good put.

Since ECN is only effective in combination with an Active Queue Management policy, the benefits of ECN depend on the precise AQM being used. ECN reduces the number of packets dropped by a TCP connection. But, effects of ECN on bulk throughput are less clear, due to the fact that modern TCP implementations are fairly good at resending dropped segments in a timely manner when the sender's window is large. Use of ECN has been found to be detrimental to performance on highly congested networks when using AQM algorithms that never drop packets. Modern AQM implementations avoid this pitfall by dropping rather than marking packets at very high load. This paper will extend to evaluation of using ECN with other version of TCP and active queue management mechanisms.

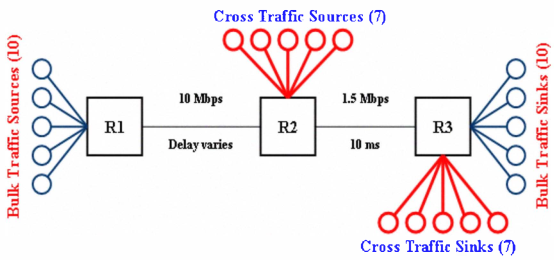


Figure 6. Network Topology for Experiment 2

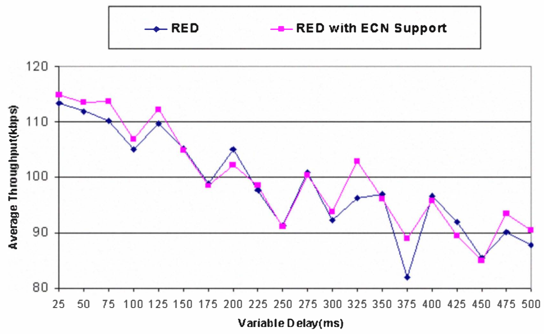


Figure 7. The average throughput of the 10 sources

TABLE II. THROUGHPUT STATISTICS FOR EXPERIMENT 2

Delay	Average Throughput	
	RED	***RED with ECN***
50	111.8859	113.5114
100	105.0292	106.8666
250	91.2531	91.0615
375	81.9883	89.0164

REFERENCES

[1] V. Jacobson and M. J. Karels. Congestion Avoidance and Control. In SIGCOMM'88, 1988.

[2] J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-Friendly Congestion Control (Extended Version)," Tech. rep. TR-2001-002, Dept. of Math. and Comp. Sci., Univ. of Mannheim, Feb. 2001.

[3] A. Mankin, "Random Drop Congestion Processing". In Proceedings of the 1990 SIGCOMM Conference, pp. 1-29. September 1990.

[4] K.Chitra, G.Padamavathi, "Classification and Performance of AQM-Based Schemes for Congestion Avoidance " , IJSCIS, Vol. 8 No. 1, April 2010.

[5] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1 N.4, August 1993, p.397-413.

[6] S. Floyd. "TCP and Explicit Congestion Notification.", Computer Communication Review, Vol. 24, No. 5:10–23, 1994.

[7] K. Ramakrishnan and S. Floyd. The addition of explicit congestion notification to IP, Sept. 2001. Internet RFC 3168.

[8] The Network Simulator - ns-2: http://www.isi.edu/nsnam/ns/.

[9] J. Padhye et al., "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," IEEE/ACM Trans. Net., vol. 8, no. 2, Apr. 2000, pp. 133–45.

[10] S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm",RFC 2582, April 1999.

[11] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, October 1996.

[12] L. S. Brakmo, L. L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet". IEEE Journal on Selected Areas in Communications (JSAC), 13(8):1465-1480, October 1995.

[13] Jinsheng Sun. King-Tim Ko.,Guanrong Chen., Sammy Chan.,Moshe sukerman., "PD –RED : To Improve Performance of RED", IEEE COMMUNICATIONS LETTER, August 2003.

[14] Shan Suthaharan, "Reduction of queue oscillation in the next generation Internet routers", Science Direct, Computer Communication, 2007.

[15] Jahoon Koo., Byunghun Song., Kwangsue Chung., Hyukjoon Lee., Hyunkook Kahng.,"MRED: A New Approach To Random Early Detection" 15th International Conference on Information Networking, February 2001.

[16] Bing Zheng ,Mogammed Atiquzzaman, "DSRED: An Active Queue Management Scheme for Next Generation Networks", Proceedings of 25th IEEE conference on Local Computer Networks LCN 2000,November 2000.

[17] D. Lin and R. Morris, "Dynamics of random early detection," in Proc. of ACM SIGCOMM '97, Cannes, France, Oct. 1997, pp. 127-137.