Global Journal of Computer Science and Technology
Volume 11 Issue 18 Version 1.0 October 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Comprehensive Analysis of Congestion Control Using Random Early Discard (RED) Queue

By Md. Abdullah Al Mamun, Momotaz Begum, Mridul Kanti Das, Md. Rubel

Dhaka University of Engineering & Technology (DUET), Gazipur, Bangladesh

Abstract - Normally all the congestion control method discard the received packet when the queue is full but it is a great problem for speed of data transfer at present. There are many ways to solve this problem. Random Early Detection (RED) algorithm is one of the most famous and powerful method to improve the performance for TCP Connection. In terms of queue management RED drops packet in considered router buffer to adjust the network traffic behavior according to the queue size. We want to investigate how high priority user datagram protocol (UDP) traffic affects the performance of lower priority Transmission Control Protocol (TCP) and proof that RED is the better for controlling the Traffic when they share the same bottleneck link with one or two classes of service.

General Terms : Network Congestion Control, TCP Variants, Network Parameters, Queuing, Drop Tail and User Data Gram Protocol.

Keywords : IETF, RED, AQM, BW, TCP Variants, NS-2, TCL and OTCL.

GJCST-F Classification : C.2.2

A COMPREHENSIVE ANALYSIS OF CONGESTION CONTROL USING RANDOM EARLY DISCARD REDQUEUE

Strictly as per the compliance and regulations of:

A Comprehensive Analysis of Congestion Control Using Random Early Discard (RED) Queue

Md. Abdullah Al Mamun[α], Momotaz Begum[Ω], Mridul Kanti Das[β], Md. Rubel[ψ]

Abstract - Normally all the congestion control method discard the received packet when the queue is full but it is a great problem for speed of data transfer at present. There are many ways to solve this problem. Random Early Detection **(RED)** algorithm is one of the most famous and powerful method to improve the performance for **TCP** Connection. In terms of queue management **RED** drops packet in considered router buffer to adjust the network traffic behavior according to the queue size. We want to investigate how high priority user datagram protocol **(UDP)** traffic affects the performance of lower priority Transmission Control Protocol **(TCP)** and proof that **RED** is the better for controlling the Traffic when they share the same bottleneck link with one or two classes of service.

General Terms : Network Congestion Control, TCP Variants, Network Parameters, Queuing, Drop Tail and User Data Gram Protocol.

Keywords : IETF, RED, AQM, BW, TCP Variants, NS-2, TCL and OTCL.

I. Introduction

A Random Early Detection (RED) is the first active queue management algorithm proposed for deployment in TCP/IP networks. The basic idea behind an active queue management algorithm is to convey congestion notification early to the TCP end points so that they can reduce their transmission rates before queue overflow and sustained packet loss occur. "It is now widely accepted that the RED controlled queue performs better than a drop-tail queue. It is an active queue management algorithm" [1]. "The tail drop algorithm, a router buffer as many packets as it can, and drops the packet when it cannot buffer. If buffers are constantly full, the network is congested" [2]. RED addresses these issues. It monitor the average queue size and drops packets based on statistical probabilities. If the buffer is almost empty, all incoming packets reaccepted. As the queue grows, the probabilities for dropping incoming packet are dropped too. RED is more fair than trail drop in the sense of it does not possess a bias against burst traffic that use only a small portion of the bandwidth. The more the more a host transmits, likely it is that packets are dropped. The most common technique of queue management is a trail drop. In this method packets are accepted as long as there is space in the buffer when it becomes full, incoming packets are dropped. This approach results in dropping large number of packets in the time congestion. This can result in lower throughput and TCP synchronization [3]. However TCP includes eleven variants (Tahoe, FullTcp, TCP/Asym, Reno, Reno/Asym, Newreno/Asym, Sack1, DelAck and Sack1/DelAck) as source and five (TCPSink, TCPSink/Asym, Sack1, DelAck and Sack1/DelAck) as destination, implementation in NS-2 [4, 5]. The base TCP has become known as TCP Tahoe. TCP Reno attaches one novel mechanism called Fast Recovery to TCP Tahoe [4]. In addition, TCP Newreno employs the most recent retransmission mechanism of TCP Reno. [6]. The use of Sacks allows the receiver to stipulate several additional data packets that have been received out-of-order within one dupack, instead of only the last in order packet received [5]. TCP Vegas offers its own distinctive retransmission and congestion control strategies. TCP Fack is Reno TCP with forward acknowledgment [7]. Transmission Control Protocol (TCP) Variants Reno, NewReno, Vegas, Fack and Sack1 are implemented in NS-2. RED supervises the average queue size and drops packets based on statistical likelihoods [3].

Author [α] : Bachelor of Science in Engineering degree from Department of Computer Science and Engineering (CSE), Dhaka University of Engineering & Technology (DUET), Gazipur, Bangladesh. PH-8801717871852. E-mail : aamcse@gmail.com

Author [Ω] : Lecturer, Department of CSE, DUET and she achieved her B. Sc. in Engineering degree from Department of Computer Science and Engineering (CSE),DUET, Bangladesh. PH-8801913184657. E-mail : momotaz03-duet@yahoo.com

Author [β] : Bachelor of Science in Engineering degree from Department of Computer Science and Engineering (CSE), Dhaka University of Engineering & Technology (DUET), Gazipur, Bangladesh. PH-+8801715543754. E-mail : mridul.duet@yahoo.com

Author [ψ] : Bachelor of Science in Engineering degree from Department of Computer Science and Engineering (CSE), Dhaka University of Engineering & Technology (DUET), Gazipur, Bangladesh. PH-+8801719972044. E-mail : rubel_cse_064056@yahoo.com

II. Random Early Detection

a) RED Parameter Setting

Average queue size *avg* is formulated [1] as:

$$avg \leftarrow (1 - wq) \times avg + w_q \times q \qquad \text{(I)}$$

Where, wq is the queue weight, q is current queue size. wq should have lower value for bustier traffic; more weight is given in this case for the historic size of the queue. As *avg* varies from *minth* to *maxth*, the packet-marking probability *pb* varies linearly from *0* to *maxp*..

$$p_b \leftarrow \frac{max_p \times (avg - min_{th})}{max_{th} - min_{th}} \qquad \text{(II)}$$

The final packet-marking probability p_a increases slowly as the count increases since the last marked packet [1]:

$$p_a \leftarrow \frac{p_b}{1 - count \times p_b} \qquad \text{(III)}$$

III. Performance Analysis of Red Model

a) Variation in Threshold Value

Table 1 : Number received packet for various TCP variants with respect to threshold for simulation time 70s

TCP variants	15	20	25	30	35
Reno	854	1185	845	711	733
Newreno	721	763	752	774	741
Vegas	821	777	685	686	625
Fack	800	721	713	644	761
Sack1	864	870	749	813	786

Table 2 : Number received packet for various TCP variant with respect to threshold for simulation time 140s

TCP variants	15	20	25	30	35
Reno	1452	1532	1333	1778	1398
Newreno	1458	1465	1501	1631	1538
Vegas	1345	1578	1350	1498	1538
Fack	1412	1754	1252	2379	1422
Sack1	1501	1339	1595	1358	1179

Table 3 : Number received packet for various TCP variants with respect to threshold for simulation time 210s

TCP variants	15	20	25	30	35
Reno	2659	2635	2376	1946	2300
Newreno	2701	2546	2032	2169	2303
Vegas	2254	2255	2301	2432	2178
Fack	2802	2462	2897	2131	2376
Sack1	2269	2416	2201	2554	2082

Table 4 : Number received packet for various TCP variant with respect to threshold for simulation time 270s

TCP variants	15	20	25	30	35
Reno	3142	3403	3312	3323	2902
Newreno	3383	3220	3204	3265	2928
Vegas	2624	2749	2778	2538	2799
Fack	3545	3088	2856	2681	4298
Sack1	3888	3216	3051	3232	3409

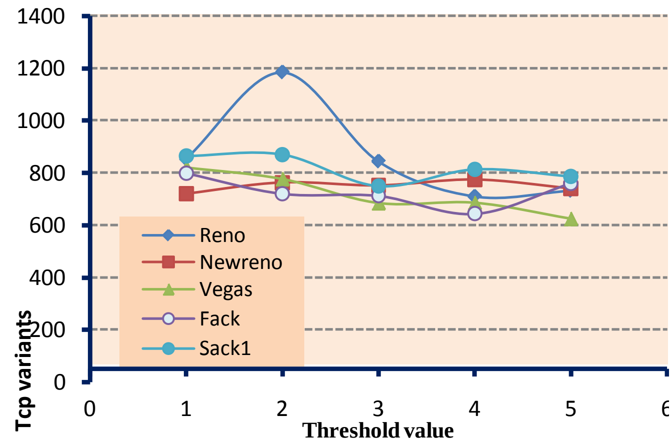

Figure1 : Graph of received packet for various TCP variants with respect to threshold for simulation time 70s

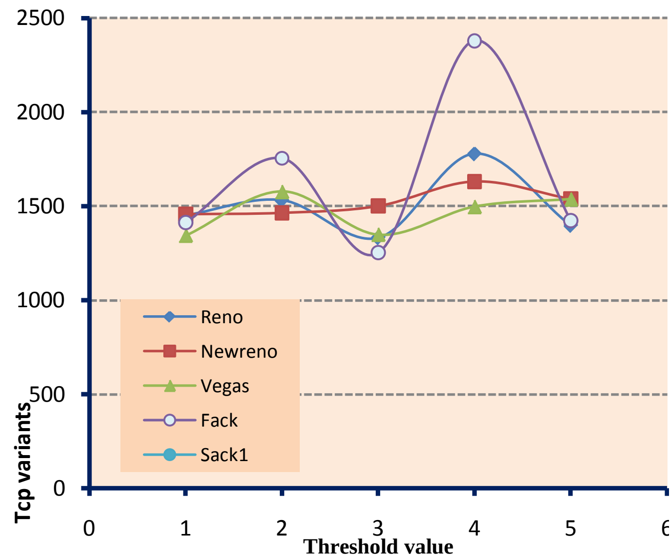

Figure2 : Graph of received packet for various TCP variants with respect to threshold for simulation time 140s

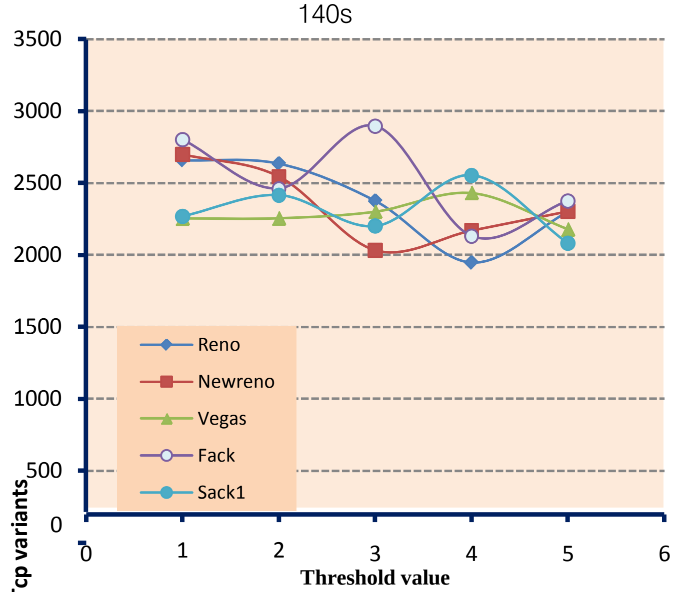

Figure3 : Graph of received packet for various TCP variants with respect to threshold for simulation time 210s

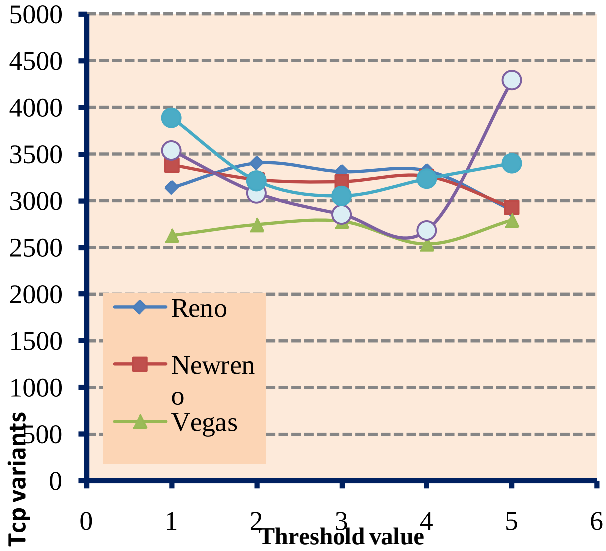

Figure4 : Graph of received packet for various TCP variants with respect to threshold for simulation time 280s

b) Performance Comparison

We that when threshold increase then variation course in received among various TCP variants and all arriving packets are received when average queue size exceeds max threshold or less than minimum threshold then packets are dropped which is shown in above all tables and corresponding figure. We found that Newreno TCP variants is the best because mean number of received packet is high mean number of dropped packet is low.

c) Comparison of TCP and UDP

i. Comparison of Received Packet

Table 5 : Comparison of received packet between UDP and TCP

Times		70s	140s	210s	280s
U	15	675	1294	1996	2586
	20	797	1222	1803	2694
D	25	758	1187	2127	2633
	30	795	1484	2085	2794
P	35	749	1336	1963	2783
T	15	566	1352	2725	2457
	20	665	1606	2374	3284
C	25	637	1438	2425	3694
	30	548	1656	2247	2832
P	35	834	1614	2413	3438

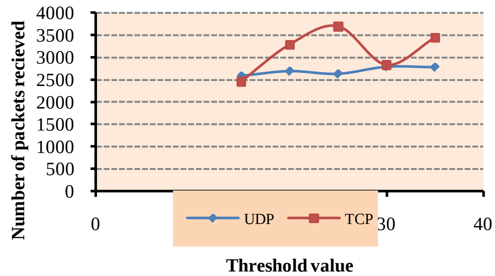

Figure5 : comparison graph of received packet between UDP and TCP for simulation time 280s

ii. Comparison of Dropped Packet

Table 6 : Comparison of received packet between UDP and TCP

Times		70s	140s	210s	280s
U	15	25	126	246	374
	20	67	104	113	354
D	25	24	53	426	696
	30	135	113	162	344
P	35	36	34	433	357
T	15	0	26	37	73
	20	0	14	33	46
C	25	0	5	36	43
	30	0	8	23	17
P	35	0	4	14	12

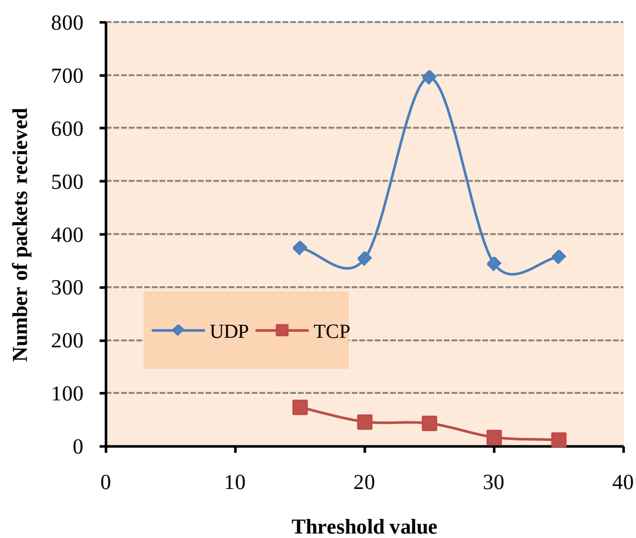

Figure6 : comparison graph of dropped packet between UDP and TCP for simulation time 280s

IV. Conclusion

From the aforementioned comparison of the performance it is found that TCP is better than UDP because packet received is higher in it with respect to UDP. That is why packet loss is lower in TCP. In case of packet drop, it is clear those packet drop is higher in UDP than TCP and also occur more congestion in it. It is possible to control congestion in TCP using RED model.

References Références Referencias

1. S. FLOYD and V. JACOBSON: „Random Early Detection gateways for congestion avoidance ‟. *IEEE/ACM Transactions on Networking*, 1993, 1 (4), pp. 397-413.
2. MD. SHOHIDUL ISLAM, MD. NIAZ MORSHED, MD. SHARIFUL ISLAM AND MEJBAHUL AZAM: "An experimental analysis of random early discard (RED) queue for congestion control", 2011.
3. S. Floyd. RED: Discussions of setting parameters. *http://www.aciri.org/floyd/REDparameters.txt*, 1997.
4. NS, The network simulator-ns-2.27 URL:http://www.isi.edu/nsnam/ns
5. "The ns Manual", The VINT Project, A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.
6. Tanja Lang, (2002), "Evaluation of different TCP versions non-wireline environments", *The University of South Australia, Institute for Telecommunications Research*.
7. Merida, Venezuela and ESSI, Sophia-Antipolis (2003), "NS Simulator for beginners", Lecture notes, 2003-2004, Univ. de Los Andes, France.
8. C.V. HOLLOT, VISHAL MISRA, DON TOWSLEY AND WEI-BO GONG: A Control Theoretic Analysis of RED. *IEEE INFOCOM 2001.*

Global Journal of Computer Science and Technology Volume XI Issue XVIII Version I October 2011