Random Early Discard (RED-AQM) Performance Analysis in Terms of TCP Variants and Network Parameters: Instability in High-Bandwidth-Delay Network

Mohammad Abu Obaida
Dept. of CSE, DUET
Gazipur-1700,
Bangladesh

Md. Sanaullah Miah
Dept. of CSE, DUET
Gazipur-1700,
Bangladesh

Md. Abu Horaira
Software Engineer
DataSoft BD Limited
Chittagong, Bangladesh

ABSTRACT
Conventional congestion control methods (e.g. DROP TAIL) discards all received packets after the queue is full moreover results in low-network performance. To address this problem, RED was proposed to improve the performance of TCP connections. As a queue management mechanism, it drops packets in the considered router buffer to adjust the network traffic behavior according to the queue size. In application, TCP Variants (Reno, NewReno, Vegas, Fack and Sack1) show oscillatory curve of packet reception if RED is considered for queuing, besides, some variants out performs in receiving packets over different network parameters that this paper analyzes and finds out. However, an increase in link capacity (with the resulting increase of per-flow bandwidth) will cause significant degradation in TCP's performance, irrespective of the queuing scheme used. Hence the network is prone to instability with the rise in the number of High-bandwidth-delay product that is also attended to in this paper.

General Terms
Network Congestion Control, TCP Variants, Network Parameters, Queuing, Drop Tail, High-bandwidth-delay and Random Early Marking.

Keywords
RED, AQM, BW, TCP Variants, NS-2.

1. INTRODUCTION
Random Early Marking puts forward a great deal of improvement for the purpose of Congestion Control rather than conventional Drop Tail mechanism does. The rationale behind the discrepancy in RED performance for different TCP Variants is that each of the Variants type possesses some special decisive factors. Such as, the base TCP has become known as TCP Tahoe. TCP Reno attaches one novel mechanism called Fast Recovery to TCP Tahoe [6]. In addition, TCP Newreno employs the most recent retransmission mechanism of TCP Reno. [7]. 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 [8]. TCP Vegas offers its own distinctive retransmission and congestion control strategies. TCP Fack is Reno TCP with forward acknowledgment [9]. 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 [2].

2. RANDOM EARLY MARKING

2.1 RED, an improvement over Drop Tail
Droptail network components discard packets when its FIFO queue is full. Under heavy load conditions, droptail routers grounds global synchronization, a phenomenon in which all senders sharing the same bottleneck router/link shut down their transmission windows at almost the same time, thereby causing a abrupt drop in the bottleneck link exploitation [10]. Besides, droptail routers are biased against bursty sources [11]. This is because, when a burst of packets from a sender arrives at a fully occupied queue, a sustained packet drop within the same window of data occurs. Floyd et al. [12] and Xu and Ansari [13] showed that TCP [14], lacks the capability to recuperate from such multiple packet losses within the same window of data. Therefore, the TCP sender has to rely on retransmission timeouts to recover the lost packets that considerably slows down the transmission rate of a TCP flow. On the Contrary, RED monitors the average queue size and drops packets based on statistical probabilities [2]. As the queue grows, the probability for dropping an incoming packet grows too. RED is shown to effectively tackle both the global synchronization problem and the problem of bias against bursty sources [1].

2.2 RED Parameter Settings
Average queue size *avg* is formulated [1] as:

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

Where, w_q is the queue weight, q is current queue size.

w_q should have lower value for burstier traffic; more weight is given in this case for the historic size of the queue.

As *avg* varies from min_{th} to max_{th}, the packet-marking probability p_b varies linearly from *0* to max_p.

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

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

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

To implement RED with NS-2 or other simulation software:

The optimal setting for min_{th} (minimum threshold) depends on exactly what the desired tradeoff is at that router between low average delay and high link utilization. The burstier the traffic arrival process, the higher min_{th} would have to be to achieve a given average link utilization.

1. NS-2 simulator allows basic RED automatic setting of parameters [8] according the expected RTT. By setting RED's parameters (*thres_ = 0, maxthres_ = 0, q_weight = -1*), this mechanism can be enabled.

2. The popular RED rule of thumb that is published in Sally Floyd's discussion on RED parameters [15]. The parameters are min_{th} =5 (ascertained or assumed), max_{th}= 3 x min_{th}, max_p= 0.1, w_q=0.002. The rules, of course, are flexible, e.g. burstier communication should have lower w_q.

3. Using fluid-flow model of TCP behavior to predict and achieve stability of the linear control system, model for setting RED parameters [3] is given by-

$$\frac{max_p}{max_{th} - min_{th}} = L_{red} \le \frac{(2N^-)^2}{(R^+C)^3} \times \sqrt{\frac{w_g^2}{K^2} + 1} \qquad \text{(IV)}$$

Where:

$$K = \frac{\log_e(1-\alpha)}{\delta} \qquad \text{(V)}$$

When the number of nodes: $N \ge N^-$.

Average RTT:

$$R_0 \leftarrow \frac{q_0}{C} + T_p \qquad \text{(VI)}$$

Maximum expected RTT: $R_0 \le R^+$

$$w_g = 0.1 \times minimum\left\{\frac{2N^-}{(R^+)^2} \times C,\ \frac{1}{R^+}\right\} \qquad \text{(VII)}$$

(w_g^{-1} is approximately the time constant of the feedback loop). L_{red} should have the highest possible value; K is usually set to 0.005.

T_p=Propagation Delays(sec)

C=Line Capacity (packets/seconds)

3. PERFORMANCE ANALYSIS

3.1 Variation in Bandwidth over Simulation

Red queuing in application with respect to various TCP variants, in a condition of varied bandwidth maintaining other parameters unaltered, number of packets received are mentioned in Table 1 (90sec simulation), Table 2 (240sec simulation) and packets received vs. bandwidth is delineated consequently in Figure 1, Figure2. As illustrated in the graph, it is apparent that some variants perform very sound under low bandwidth conditions, however, other shows steady or improved performance for a range of bandwidths.

Table 1. Total number of packet received for various TCP variants with respect to Bandwidth for simulation time 90s.

TCP Variants	Bandwidth				
	2Mb	4Mb	8Mb	16Mb	32Mb
Reno	2264	2050	1986	2000	1916
NewReno	2303	1944	2246	2048	2013
Vegas	1833	1738	2098	1573	2011
Fack	1938	1963	2187	2222	2166
Sack1	2314	1963	1849	2175	1989

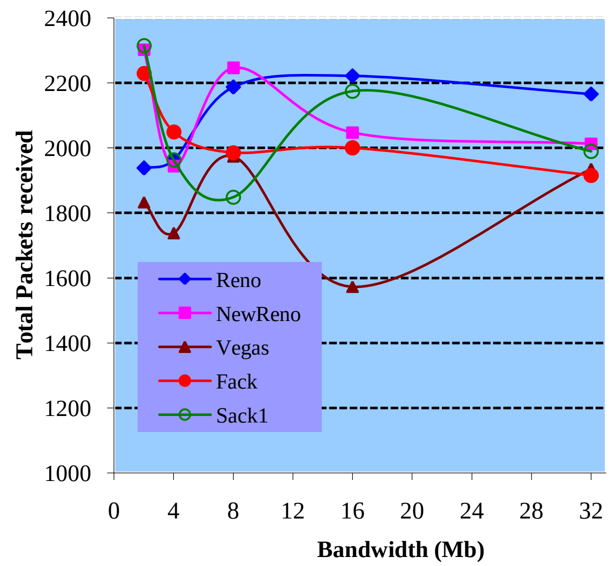

Figure 1.Graph of total packets received for various TCP Variants with respect to Bandwidth for simulation of 90s.

Table 2. Total number of packet received for various TCP variants with respect to Bandwidth for simulation of 240s

TCP Variants	Bandwidth				
	2Mb	4Mb	8Mb	16Mb	32Mb
Reno	5889	5234	5476	5975	5648
NewReno	6012	5469	5349	5895	5648
Vegas	4774	4833	4972	4757	5249
Fack	5355	5663	5605	5603	5836
Sack1	5782	5077	5624	5603	5170

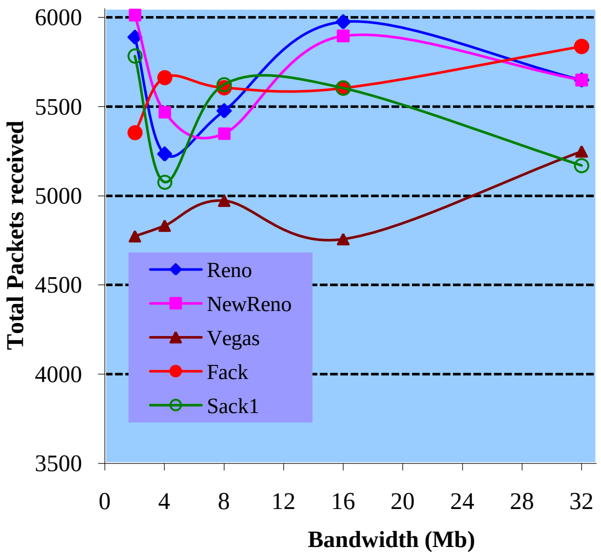

Figure 2.Graph of total packets received for various TCP Variants with respect to Bandwidth for 240s simulation.

3.2 Variation in Delay over Simulation

Some of the TCP Variants are indifferent to the variation in delay; on the contrary, other variant seems extremely sensitive to the alteration in the delay. Red queuing in application with respect to various TCP variants number of packets received are mentioned in Table 3 (90sec simulation), Table 4 (240sec simulation) and depicted accordingly in Figure 3, Figure 4. As demonstrated in the graph, easily noticed that some variants operates exceptionally well under low bandwidth settings, however, other shows stable or improved performance for a band of bandwidths.

Table 3. Total number of packet received for various TCP variants with respect to Delay for simulation of 90s.

TCP Variants	Delay					
	1ms	5ms	10ms	20ms	50ms	100ms
Reno	2326	2106	2230	1888	2133	1967
NewReno	1838	1976	2303	2356	2134	2120
Vegas	1860	1770	1833	1990	1889	2089
Fack	1997	1767	1938	2023	1970	2108
Sack1	2029	1949	2314	1860	1947	1813

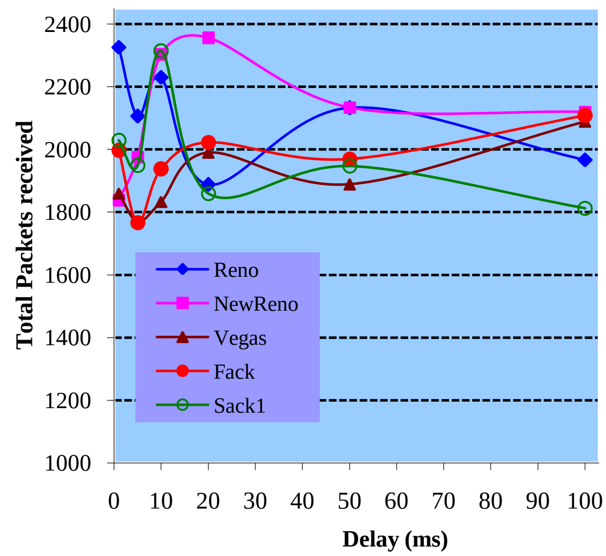

Figure 3.Graph of total packets received for various TCP Variants with respect to Delay for simulation of 90s.

Table 4. Total number of packet received for various TCP variants with respect to Delay for simulation of 240s.

TCP Variants	Delay					
	1ms	5ms	10ms	20ms	50ms	100ms
Reno	5702	5329	5889	5056	5456	5636
NewReno	5700	5536	6012	5987	5170	5381
Vegas	4763	4849	4774	4855	4554	4894
Fack	5984	5595	5355	5954	5241	5407
Sack1	5607	5664	5782	5775	5561	5580

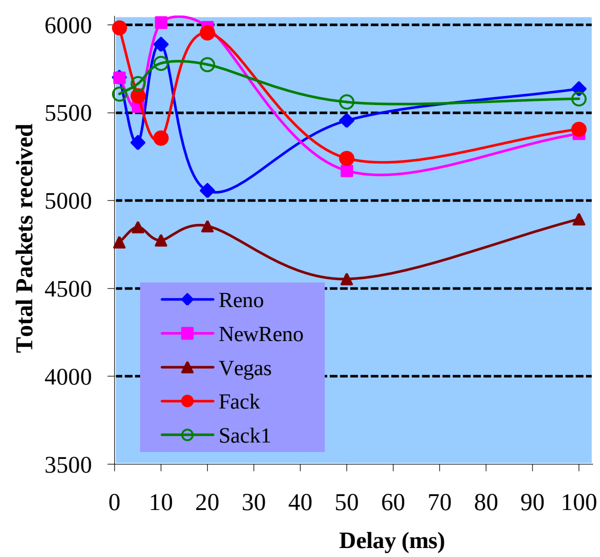

Figure 4.Graph of total packets received for various TCP Variants with respect to Delay for simulation of 240s.

3.3 Rating the TCP Variants

To deduce, a deliberate scrutiny of the statistics and graphs demonstrated above advocate the following findings and decisions for RED queuing-

Variation in Bandwidth over Simulation (3.1):

Reno, NewReno, Fack→ Receive more packets and steady.
NewReno, Sack1, Reno→ Excellent performance in low BW.
Vegas, Sack1, Newreno→Unstable over variation in BW.
=>Most consistent and better performance literally: *Reno*.

Variation in Delay over Simulation (3.2):

NewReno, Reno, Sack1→ Receive more packets.
Reno, Fack, Sack1→Superior performance in low Delay.
Vegas, Sack1→Unstable over variation in delay.
Sack1, Vegas, Fack→Least delay sensitive.
=>Most regular and better performance literally: *NewReno*.

3.4 Responses in High-Bandwidth-Delay Network:

Irrespective of the queuing scheme used, an increase in link capacity (with the ensuing increase of per-flow bandwidth) will cause considerable degradation in TCP's performance [5].By casting the problem into a control theory framework, Low et al. [16] show that as capacity or delay increases, Random Early Discard (RED) [1], Random Early Marking (REM) [17] or any other Active Queue Management scheme (AQM) all eventually become oscillatory and prone to instability. In simulation, if iterated for several times without any alteration in RED and network parameters; certain fluctuation is noticed for several TCP variants. Thereby, confirming the unsteadiness of RED queuing with the rise in the number of High-bandwidth-delay products in the network. Table 5, reveals some of such aberrations.

Table 5. Packet received alongside several Parameter Settings of network.

	Network Parameter Settings	Observations		
	RED Queue Parameter: min_thresh:18, thresh:40, queue:50	1	2	3
reno	Bandwidth:4MB Delay:10ms, Time:90sec	1806	2478	1963
reno	Bandwidth:2MB Delay:1ms, Time:90sec	2050	3059	2326
reno	Bandwidth:2MB Delay:5ms, Time:90sec	1953	3249	2106
reno	Bandwidth:2MB Delay:5ms, Time:240sec(**)	4747	6305	5316
newreno	Bandwidth:2MB Delay:10ms, Time:90sec	1698	2948	2263
newreno	Bandwidth:4MB Delay:10ms, Time:240sec	4903	6124	5370
newreno	Bandwidth:32MB Delay:10ms, Time:240sec(**)	5282	8098	5648
vegas	Bandwidth:32MB Delay:10ms, Time:90sec	1740	2359	1935
vegas	Bandwidth:2MB Delay:5ms, Time:90sec	1315	2197	1800
vegas	Bandwidth:2MB Delay:100ms, Time:240sec(**)	4275	5607	4801
fack1	Bandwidth:32MB Delay:10ms, Time:90sec	1847	2612	1916
fack1	Bandwidth: 2MB Delay:10ms, Time:240sec	5355	5061	6297
fack1	Bandwidth:32MB Delay:10ms, Time:240sec(**)	5655	4947	6908
fack1	Bandwidth:2MB Delay:5ms, Time:90sec	1767	1540	2790
fack1	Bandwidth:2MB Delay:1ms, Time:240sec	5121	6840	5993
fack1	Bandwidth:2MB Delay:20ms, Time:240sec	5954	6835	5121
sack1	Bandwidth:2MB Delay:10ms, Time: 90sec	2275	2314	3250
sack1	Bandwidth:8MB Delay:10ms, Time: 90sec	1653	1849	2335
sack1	Bandwidth:2MB Delay:10ms, Time: 240sec(**)	6624	5188	5534
sack1	Bandwidth:2MB Delay:1ms, Time: 240sec	4950	5641	6232

(Table 5. Continued…)

(**) marked observations of 240s simulation are portrayed in the graph of Figure 5 (An aberrant observation/Variant).

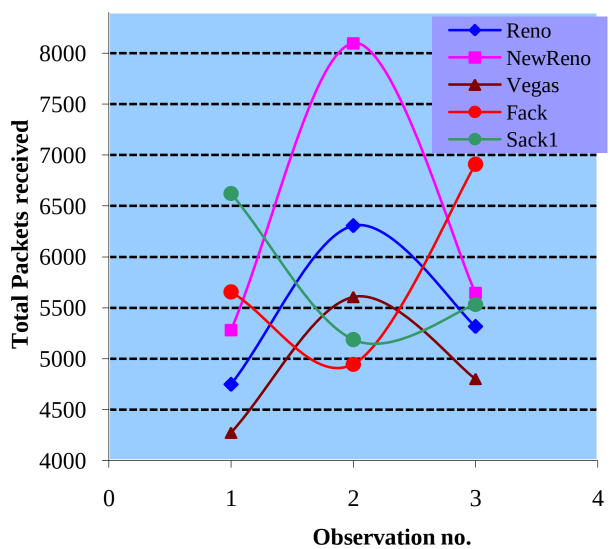

Figure 5.Graph of total packets received for various TCP Variants under the condition of unsteadiness for 240s.

As depicted in the Figure 5, we discern easily, performances of RED queuing in case of high-bandwidth-delay networks is undulating and unsteady to a great extent.

4. CONCLUSION

For the Internet to flourish incessantly its congestion control mechanism must remain efficient and effective as the network gestates. Our extensive simulations illustrate that some of the TCP Variants maintains good utilization and steadiness over a band of bandwidths or a scale of delays. However, RED is understood to sustain a network steady to a definite limit in network parameter's settings, in addition fails to deal with network extremes. To deal with both steady-state and dynamic environments (with web-like traffic and with impulse loads), a new mechanism XCP is proposed that explicitly deals all sort of congestion problems even for high-bandwidth-delay networks maintaining stability and implying fairness in a better manner at all aspect than any other queuing mechanism does.

5. REFERENCES

[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] S. Floyd and V. Jacobson. The synchronization of periodic routing messages. *IEEE/ACM Transactions on Networking*, 2(2):122{136, April 1994.

[3] C.V. HOLLOT, VISHAL MISRA, DON TOWSLEY AND WEI-BO GONG: A Control Theoretic Analysis of RED. *IEEE INFOCOM 2001*

[4] YIGUANG HONG, YONGHAN CAO, HAIRONG SUN, KISHOR S. TRIVEDI: 'RED parameters and Performance of TCP Connections'. *In part by ARO-MURI "Mathematics of Failures in Complex Systems"*.

[5] DINA KATABI, MARK HANDLEY, CHARLIE ROHRS: Congestion Control for High Bandwidth-Delay Product.

[6] NS, The network simulator-ns-2.27. URL:http://www.isi.edu/nsnam/ns

[7] Tanja Lang, (2002), "Evaluation of different TCP versions in non-wireline environments", *The University of South Australia, Institute for Telecommunications Research*.

[8] "The ns Manual", The VINT Project, A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.

[9] Merida, Venezuela and ESSI, Sophia-Antipolis (2003), "NS Simulator for beginners", Lecture notes, 2003-2004, Univ. de Los Andes, France.

[10] L. Zhang, S. Shenker, and D.D. Clark, Observations on the dynamics of congestion control algorithm: the effects of two way traffic, *Proceedings of the Conference on Communications Architecture &Protocols*, 133-147, Zurich, Switzerland, 1991.

[11] R. Mahajan, S. Floyd, and D. Wetherall, Controlling High Bandwidth Flows at the Congested Router, *Proceedings of the Ninth International Conference on Network Protocols*, 192, November 11-14, 2001.

[12] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, An Extension to the Selective Acknowledgement (SACK) *Option for TCP, RFC Editor*, 2000.

[13] K. Xu, and N. Ansari, Stability and fairness of rate estimation based AIAD congestion control in TCP, *IEEE Communications Letters*, 9, 378-380, 2005.

[14] Information Science Institute, Transmission Control Protocol, *in: RFC793*, 1981.

[15] S. Floyd. RED: Discussions of setting parameters. *http://www.aciri.org/ floyd/REDparameters.txt*, 1997.

[16] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle. Dynamics of tcp/aqm and a scalable control. *In Proc. of IEEE INFOCOM*, June 2002.

[17] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. Rem: Active queue management. *IEEE Network*, 2001

Authors Profile:

Mohammad Abu Obaida[1] obtained his Bacholor of Science in Engineering degree from Department of Computer Science and Engineering (CSE), Dhaka University of Engineering & Technology (DUET), Gazipur, Bangladesh in 2011. At present he is performing extensive research on networks and cryptography. His key research interest includes Cryptography, Network and Web Security, Wireless networking, Software Architecture, Machine Vision, Artificial Intelligence, Protocol analysis and Algorithm design.

Md. Sanaullah Miah achieved his B. Sc. in Engineering degree from Department of Computer Science and Engineering (CSE), DUET, Bangladesh in 2011. Being enthusiastic about mobility management in IPV6, he is currently performing extensive research on the field. Artificial Intelligence, Wireless networking, Cyber Security and Software Engineering are the major research interests to him.

Md. Abu Horaira who serves as a Software engineer to DataSoft Systems Bangladesh Limited, completed Bachelor of Science in Engineering degree from Department of CSE, DUET, Gazipur, Bangladesh in 2009. Pattern Recognition, Artificial Intelligence, System Architecture, Neural and Web Networks are the key subject of interest to him. In addition, Horaira et. al. developed a complete OCR for any local language.