ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (201-205), Month: October - December 2015, Available at: www.researchpublish.com

# Comparative Study on AQM Mechanisms for Internet Congestion Control

Ch. Ramadevi[1], N. Vyaghreswara Rao[2]

[1]Assistant professor, Department of CSE, SVEC, Tirupati, Andrapradesh, India

[2] Professor, Department of CSE, CVR college of Engineering, Hyderabad, A.P

*Abstract:* **Internet congestion occurs when the aggregate demand for a resource exceeds the available capacity of the resource. At times of high congestion, the end user experiences not only very poor performance, long delays in data delivery, wasted resources due to lost or dropped packets but also congestion collapse in which all communication in the entire network ceases. The things that make the congestion problem more difficult, the architectural approaches for AQM and its variants along with new efficient mechanism Adaptive Enhanced Random Early Detection (AE-RED) were discussed.**

*Keywords:* **Congestion Control, AQM, RED, AE-RED.**

## 1. INTRODUCTION

The rapid growth of the Internet and today's increased demand to use the Internet technologies for wide variety of the applications are mostly time-sensitive video and voice applications which requires the user's to design and implement the new Internet Architectures. So, in order to maintain good network performance, certain mechanisms must be provided to prevent the network from being congested for any significant period of time. Basically, there are two approaches to handle congestion are congestion control and congestion avoidance. The first one, is comes into play after the network is overloaded, i.e., congestion is detected. And the second form is comes into play before the network becomes overloaded, i.e., congestion is expected. Therefore, congestion becomes the big challenge in today's increasing Internet traffic. And this is caused by a shortage of buffer space, slow links, slow processors and packet losses. In current TCP/IP networks, TCP packet loss indicated either by timeout or triple duplicated acknowledgment [1], is used as an indication as network congestion. Then, the TCP controls its sending rate by limiting its congestion window size. There are many congestion control approaches have been proposed. First, the network algorithms such as active queue management (AQM), executed by network components such as routers, detect network congestion, packet losses, or incipient congestion, and inform traffic sources. And second, source algorithms adjust the source's data sending rate into the network. The basic design issues are what to feedback from network algorithms and how to react from source algorithms. This paper is organized as follows: First, we describe existing AQM algorithms, RED and its variants in additions to the problems associated with their design and implementation. Second, we discuss problems associated with AQM proposals and the proposed mechanism EE-RED. Finally, survey on architectural approaches such as source or network algorithms modifications and economic modifications including pricing or optimization of allocated resources. And this paper ends with conclusions.

## 2. ACTIVE QUEUE MANAGEMENT MECHANISMS: RED AND ITS VARIANTS

Due to the massive rapid growth in Internet propagation and with widespread use of TCP/IP, the congestion control mechanism becomes the decisive factor in improving the efficiency of TCP. RFC 793 is the first TCP version with its basic configuration based on window-based flow control. TCP Tahoe is the second generation of TCP version which includes two new techniques, congestion avoidance and fast transmission. TCP Reno is the third version of the first developed series, and it is standardized in RFC 2011, where the congestion control mechanism is further extended by fast recovery algorithm. All these TCP variants detect [2-3] congestion only after a buffer at an intermediate router has already overloaded and packets have been lost. To cope with the bursty traffic effectively, the intelligent congestion control mechanism is Tail Drop. And this is based FIFO based queue management algorithm. To overcome the drawbacks of lockout and full queue phenomenon the solution is to drop packets before a queue becomes full so that a source can respond to congestion before buffers overflow. This approach is called active queue management (AQM). And RED is the example of this approach. The basic design objectives of RED are to minimize packet loss, queuing delay, maintain high link utilization, and remove biases against bursty sources and also global synchronization. In order to achieve these goals RED introduces an enhanced control mechanism involving randomized packet dropping and queue length averaging. RED and its variants are as follows:

**Random Early Detection (RED):** Proposed by Sally Floyd and Van Jacobson [4] in 1993, it calculates average queue size by using a low pass filter with Exponential Weighted Moving Average (EWMA). Routers and gateways using RED signals congestion to TCP by drop packets probabilistically before queue becomes full and this drop probability is depends on average queue size ($q_a$). If this is less than $min_{th}$ the packet is allowed and if this is in between $min_{th}$ and $max_{th}$ , the packet is marked or dropped with some probability Pa. If the $q_a$ is greater than $max_{th}$ then all incoming packets are dropped.

$$q_a = (1-w_Q) \times q_a + w_Q \times Q \qquad (1)$$

$$P_a = max_p \times (q_a - min_{th})/max_{th} - min_{th} \qquad (2)$$

Many AQM based approaches such as ARED[5],Proportional Integral Controller Algorithm[6], DRED[7],SRED[8],BLUE [9],Weighted FRED[10],GRED[11], REM[12] ,FQA[13] and AVQ[14 ] have been proposed, and their performance has been evaluated [15].

Adaptive RED: It attempts to maintain suitable operating parameters in RED by dynamically adjusting $max_p$ in eq.2 based on observed queue length dynamics. ARED increases $max_p$ when $q_a$ exceeds $max_{th}$ and decreases $max_p$ when $q_a$ goes below $min_{th}$. ARED also work on not go underneath a packet loss probability of 1%, and it should not exceed a packet loss probability of 50%

**Proportional Integral Controller Algorithm**: To overcome the limitations of response speed, stability, coupling between queue length and loss probability of RED this was designed as given below. It can be implemented at a RED router as :

$P=a* (q-q_{ref})-b * (q_{old}-q_{ref}) + p_{old}; p_{old}-p; q_{old}=q;$

Where q is the present queue length, $q_{ref}$ is a desired queue length and a, b are constants.

**Dynamic RED**: It attempts to maintain the EWMA queue length close to a desired queue length, $Q_{ref}$, to stabilize the utilization around a pre-defined level. DRED adjusts the packet drop probability based on the deviation of the queue length from $Q_{ref}$.

**Stabilized RED**: SRED works to stabilize the queue size at a level independent of the number of active flows, the drop probability of packets is computed by obtaining the active flow to adjust instant queue size. SRED maintain its own virtual cache which like a container called a zombie list. The zombie list stores both source and destination addresses for each arriving packet. When the zombie gets its full rate a random zombie is ejected from the list and compared with the source and destination addresses for the new packet. If it belongs to the same flow is set to one. Otherwise, it is set to zero, and with a certain probability P, the content of virtual list may be replaced by the source and destination of this new packet.

**BLUE:** Developed by Wu-Chang and Feng et al. which uses packet los and link idle events rather than the queue length to control congestion. This increased the packet drop probability in response to a buffer overflow and decreases the packet drop probability when the link becomes idle.

**Flow RED**: This algorithm was developed by Lin and Morris; it continuously monitors the current queue occupancy by a given flow. FRED is interested in keeping the state flow information. So it penalizes non adaptive connection by imposing a maximum number of buffered packets; FRED depends on calculating the average queue size at both packet arrival and departure. FRED is protecting week flows by accepting packets from low bandwidth flows. FRED provides fair sharing for large numbers of connection by accepting two-packet-buffer FRED obtains two parameters ($min_q$) and ($max_q$) which refer to both minimum and maximum number of packets gets from each flow to the buffer. FRED computes the average per-active-flow buffer usage by maintaining a global variable ($avg_{cq}$). It maintains the number of active flow and when the flow is active ($Q_{len}$>$max_q$) it maintains a count of buffer packets ($Q_{len}$), and a count of times.

**Gentle RED**: Sally Floyd, 2000 was proposed in order to increase throughput and reduce the undesired oscillation in buffer size of router by enhancing parameter settings of RED. GRED was evaluated using same simulation as it used in RED. Packet loss rate for GRED is less than the RED because a new parameter was introduced which Double $max_{th}$

**Random Exponential marking algorithm:** Snajeewa Athuraliya et al. proposed this algorithm with the aim to achieve high link utilization, negligible packet loss and end-to-end delay and scalability. REM embodies a mechanism for the precise communication of link congestion prices, so that the link congestion state variable is exactly the congestion price as in the utility optimization. A REM link marks a packet at link l with a probability based on the link price pl state, and a global encoding constant $\phi$ ($1 < \phi$).

**Weighted Fair Queuing algorithm**: This is mainly used in the multimedia integrated services. Weighted fair queuing is a data packet scheduling technique allowing different scheduling priorities to statistically multiplexed data flows. Weighted fair queuing is popular because it approximates generalized processor sharing to within one packet transmission time, regardless of the arrival patterns.

**Adaptive Virtual Queue**: This algorithm uses a modified token bucket models a virtual queue to regulate buffer utilization rather than the queue length. AVQ adjusts the size and link capacity of the VQ proportional to the measured input rate and drops packets when the VQ overflows. In addition, since RED is an AQM algorithm focused on TCP/IP based best effort services, some variants also have been proposed and evaluated for differentiated services[16], multimedia[17], and ATM traffic control[18].

## 3. PROBLEMS ASSOCIATED WITH EXISTING AQM MECHANISMS

To control bursty Internet traffic effectively and efficiently AQM proposals recommend using the average queue length as a congestion indicator. Mismatch between Macroscopic and Microscopic behavior of Queue Length: AQM algorithms try to avoid congestion, stabilize queue dynamics and maintain low end-to-end delay by controlling the average queue length at a router. And we call the long-term dynamics of the average queue length as macroscopic behavior of a router. Conversely, the short-term behavior of actual queue length as microscopic behavior of a router. The studies show the different queue length dynamics between the actual queue and the average queue [8, 19].

Insensitivity to the Input Traffic Load Variation: AQM congestion control algorithms take only the buffer utilization into account as a measure of the severity of congestion. In RED [4], the router uses only the average queue length as a congestion indicator and average queue length is insensitive to the input traffic load variation, the router cannot detect incipient congestion effectively. This insensitivity of the congestion indicator to the input traffic load variation causes a fairness problem of packet dropping among connections.

Configuration Problem: AQM parameter configuration is a difficult design task. Many modified AQMs have been proposed, in which they have shown better performance by means of analytic modeling and simulation study. And each modified AQM proposal is good only for some particular traffic conditions, neither for realistic IP traffic nor a heterogeneous traffic environment.

## 4. DEVELOPING THE PROPOSED METHOD

The proposed Adaptive Enhanced-RED (AE-RED) mechanism is mainly aimed to drop connection packets at severe congestion adaptively by considering the network conditions. The basic idea of this mechanism is to improve the throughput, maximize the queue size and resource utilization at time of network congestion.

The proposed algorithm works as follows

1. Construction of node creation
2. Binding up of created nodes with application

3. Develop the proposed method along with existing RED algorithm
4. Comparing the results and make performance evaluations.

The algorithm depends on enhancement of the average queue size on a way that limits queue size to minimize the delay and packet loss rate as compared to RED.The AE-RED works to make the queue more stable. Average queue size calculation is taking place in the low pass filter in an exponential weighted moving average. The average queue size is calculated according to the *following algorithm for AE-RED:*

*target = ( $max_{th}$ + $min_{th}$) / 2;*

*Every $q_{avg}$ update:*

*For each arrival packet before the buffer overflow*

*if ( $q_{avg}$< q(size) < critical(th) )*

*$q_t$ =q –target;*

*$q_{avg}$= $q_t$ (1-$w_q$) + q. (qt -$w_q$) ;*

It depends on the queue weight parameter ($w_q$) (i.e., the queue weight is determined by the size and duration of bursts in queue size that are allowed at the gateway) considering the time constant of the low pass filter. The AERED take a new parameter beside the $w_q$ which is called target queue ($q_t$) (i.e., the difference between the current queue size and the average of the maximum threshold and minimum threshold.

| **SL.NO** | **Congestion control approach** | **Congestion measure** | **Performance metric** |
|---|---|---|---|
| 1 | RED | Average Queue length | Global synchronization, global power,Fairness, parameter sensitivity, packet loss and queuing delay |
| 2 | REM | Average queue length + input rate | Goodput , packet loss, fairness, buffer occupancy |
| **3** | **DRED** | **Instantaneous queue length** | **Link utilization , packet loss rate** |
| **4** | **CHOKe** | **Average queue len**gth | **Stateless,easy to implement , minimum overhead, trou**ghput |
| **5** | **AVQ** | **Input rate** | **Packet losses, link utilization, responsiveness to changin**g network conditions, queue length |
| **6** | **SFB** | **Packet loss + link utilization** | **Throu**ghput, delay, queue lengh |
| **7** | **E-RED** | **Queue len**gth | **Queue len**gth, throughput, stability, queuing delay |
| **8** | **LRED (proportional controller)** | **Instantaneous queue len**gth + packet loss ratio | **Response time, robustness, flexible system confi**gurations |

**Fig1: Comparison of various congestion control algorithms**

## 5. CONCLUSION

This paper presented an overview about the congestion control mechanism and concentrate on the RED algorithm and its variants and their important roles in congestion avoidance, including our proposed algorithm. It has proposed an enhancement to existing RED algorithm called AERED which does not require modification to end system. This scheme helps to reduce the average queue size of the RED queue. AERED results in small queue size which leads to less delay and low packet loss rate.

## REFERENCES

[1] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM '88*, 1988, pp. 314–29.

[2] G. A. Abed, M. Ismail, and K. Jumari, "Appraisal of Long Term Evolution System with Diversified TCP's," *Modelling Symposium (AMS), 2011 Fifth Asia*, 2011, pp. 236-239.

[3] G. A. Abed, M. Ismail, and K. Jumari, "Traffic Modeling of LTE Mobile Broadband Network Based on NS-2 Simulator," *Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on*, 2011, pp. 120-125.

[4] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Net.*, vol. 1, no. 4, Aug. 1993, pp. 397–413.

[5] W. Feng *et al.*, "A Self-Configuring RED gateway," *Proc. INFOCOM '99,* 1999.

[6] C. Hollot, V. Misra, D. Towsley, and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. *IEEE INFOCOM*, 2001.

[7] J. Aweya, M. Ouellette, and D. Y. Montuno, "A Control Theoretic Approach to Active Queue Management," *Computer Networks*, vol. 36, no. 2–3, 2001, pp. 203–35.

[8] T. J. Ott, T. V. Lakshman, and L. Wong, "SRED: Stabilized RED," *Proc. IEEE INFOCOM '99*, 1999, ftp://ftp.telcordia.com/pub/tjo/SRED.ps

[9] W. Feng *et al.*, "Blue: A New Class of Active Queue Management Algorithms," Technical Report CSE-TR-387-99, University of Michigan, 1999.

[10] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," Technical Report, UIUC, Feb. 2001.

[11] D. Lin and R. Morris. Dynamics of Random Early Detection. *Proceedings of SIGCOMM*, Sep 1997.

[12] S.Floyd, "Recommendations on using the gentle variant of RED," Notes, March, 2000.

[13] Sanjeewa Athuraliya, Steven H. Low, Victor H. Li and Qinghe Yin. REM: Active Queue Management. IEEE Network, 2001.

[14] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 1–12, Sep 1989. Austin,Texas.

[15] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," Technical Report, UIUC, Feb. 2001.

[16] D. D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," *IEEE/ACM Trans. Net.*, vol. 6, no. 4, Aug. 1998, pp. 362–73.

[17] M. Parris, K. Jeffay, and F. D. Smith, "Lightweight Active Router-Queue Management for Multimedia Networking," *Proc. Multimedia Computing and Net.*, vol. 3654, Jan. 1999, pp. 162–74.

[18] R. Rosolen, O. Bonaventure, and G. Leduc, "A RED Discard Strategy for ATM Networks and its Performance Evaluation with TCP/IP Traffic," *ACM Comp. Commun. Review*, vol. 29, no. 3, July 1999.

[19] M. May *et al.*, "Influence of Active Queue Parameters on Aggregate Traffic Performance," Technical Report no. 3995, INRIA, Sophia Antipolis, France, 2000, http://www.inria.fr/ RRRT/RR-3995.html.