A survey on Red and Some It’s Varients Incongestioncontrol Mechanism

Dr. Vinodani Katiyar¹, Anamol Chand Jain²
¹Dean, Faculty of Computer Science & Engineering, SRMU, Lucknow, INDIA
²Research Scholar, Department of Computer Application, TMU Moradabad, INDIA

ABSTRACT

The present paper is a survey of random early detection with some of its variants (RED, GRED, AGRED, and DGRED) for congestion avoidance mechanism. The study is conducted based on delay, throughput, packet loss and average queue length, with the aim to improve the network performance.

Keywords-- Congestion control, Gentle random early detection (GRED), Adaptive GRED, Dynamic Random early detection (DRED), DGRED, Performance evaluation.

I. INTRODUCTION

The high reliable exchange of data using the Internet has been important for its explosive growth and utilization. The Transmission Control Protocol, which is called TCP, is well known as this exchange. Under TCP, a windowflow-control mechanism is used to set its transmission rate. In this mechanism, TCP increases the window size during successful data transmission. Conversely TCP cuts the window size in half whenever a data does not reach the receiver. Such data losses called “packet losses” can affect network performance. One of causes of this is that TCP has no information of network mechanisms contributing to packet loss [1].

Some kinds of AQM schemes are proposed, e.g. Random Early Detection(RED) [2], Virtual Queue [3], Random Early Marking(REM), Adaptive Virtual Queue(AVQ) [4] and Proportional Integral Controller [5]. Based on the control theory, it seems possible to design that congestion controllers(AQM schemes) achieve better performances than those AQM schemes do. AQM design problems are important and become useful in future researches because AQM is embedded in the router having much information about circumstances of current networks.

The drawbacks of congestions are as follows. Congestion plays a major role in worsening network performance by increasing the packet dropping probability (Dp) and increasing packet loss probability (PL). In addition, congestion may lead to an increase in the mean queue length (mql) and the mean waitingtime (D) of packets, which will finally degrade the amount of packets passing through the buffer of the routers, namely, the throughput (T) [6].

Congestion is associated with the status of the average queue length (aql) which in turn affects network performance. When aql value increases, T value likewise increases. At the same time, D and PL increase, and the router buffer overflows. By contrast, when the aql value relatively decreases, D and T likewise decrease. Network efficiency is decreased in both cases. Thus, congestion control is required to maintain a stable aql value, optimize the utilization of network resources, and enhance its performance. Enormous congestion control algorithms, such as Random Early Detection(RED) [7], Gentle RED (GRED) [8], Adaptive Gentle RED (AGRED) [9], and DGRED[10] have been proposed.

II. RED, GRED, AGRED and DGRED

Enormous algorithms for congestion control, such as RED, GRED, AGRED, and other time-discrete queue analytical models, have been built based on AQM. Generally, RED detects the congestion by initially computing the aql and comparing it with the minthreshold and maxthreshold. Congestion does not occur when aql is smaller than the minthreshold. Therouters, therefore, do not drop any packet. If the aql is between the two thresholds, the arriving packet is dropped and the probability is calculated asDp to alleviate congestion. Finally, when the aql is above the maxthreshold, all arriving packets are dropped to a Dp value equal to one. Generally, RED’s drawback is the varying aql computed according to the congestion status. Hence, if the congestion status is light, the aql value will be close to the minthreshold. If the congestion status is severe, the aql value will be close to the maxthreshold; thus, the packet Dp will increase, and the buffer will overflow. Another drawback is the reliance of the computed aql on the traffic load (number of connections).
Fig. 1 Single router buffer for RED

If the traffic load is high, the aql value may exceed the maxthreshold. In such a case, network performance in many aspects will worsen. Therefore, the router buffer will drop every arriving packet. Thus, the RED parameters must be set at particular values to ensure satisfactory performance. If the traffic load is low, the aql will normally be lesser than the minthreshold. Consequently, no arriving packet is dropped. Overall, RED cannot stabilize its aql value between the minthreshold and maxthreshold when the traffic load changes suddenly (i.e., bursty traffic) [11, 12].

GRED was proposed to overcome some of the limitations in RED [12, 13]. Similar to RED, the GRED algorithm aims to manage and control the congestion networks at the early stage. GRED implements its algorithm by stabilizing the aql at a certain level. GRED employs a similar approach used by RED in calculating the Dp. However, GRED utilizes three thresholds, namely, minimum, maximum, and double maximum.

GRED also has some limitations. First, GRED deals with several threshold values. Second, GRED must set its parameters to specific values to obtain satisfactory performance (i.e., parameterization). Third, when the aql is less than the minthreshold and heavy congestion occurs, the aql will take time to adjust, during which the router buffer will likely overflow. Thus, no packets are dropped despite the overflowing GRED router buffer. The AGRED algorithm is proposed to improve the performance of GRED during router buffer congestion (i.e., deriving better quality results with reference to the mql, D, and PL performance measures). In addition, the AGRED algorithm aims to enhance the parameter settings (e.g., the maxthreshold and the maximum value of Dinit, which is the Dmax of GRED). The calculation of the aql in AGRED is also similar to that in GRED.

Fig. 2 Single router buffer for GRED AND AGRED

Therefore, AGRED decides on packet dropping in a manner similar to that in GRED. The main difference between the GRED and the AGRED lies in the calculation of the Dinit value (the initial packet Dp). In AGRED, the Dinit value varies between the Dmax values to 0.5, as long as the aql value is between the maxthreshold and double maxthreshold. In GRED, when the aql value is between the maxthreshold and the double maxthreshold, the calculated Dinit value of GRED varies from the Dmax value to 1.0.

DGRED is an extension of GRED. DGRED employs a dynamic maxthreshold.
and doublemaxthreshold to control the congestion in the router buffer at the early stage before it overflows. In DGRED algorithm a new defined value called Target aql(Taql) is calculated between the minthreshold and maxthreshold which provides better performance results. These better performance results are represented by the results of mean queue length, average queuing delay and packet loss probability when heavy congestion has occurred. DGRED also updates the maxthreshold and doublemaxthreshold parameters at the router buffer to enhance network performance. DGRED uses the GRED algorithm’s policy in dropping packets with probability when the aql is between the minthreshold and doublemaxthreshold [14].

The throughput of the compared algorithms give similar T results, whether the probability of packet arrival is set to a value lower or higher than the probability of packet departure value. On the contrary, the T results for all compared algorithms are stabilized at the value of the packet departure probability when there is congestion at the router buffer of the algorithms.

The performance measure results of PL and DP are computed after the system reaches a steady state for RED, GRED, AGRED and DGRED. The results of PL and DP are obtained as before by running the algorithm simulations ten times with various random speeds, then taking the mean of the ten results. When the packet arrival probability is smaller than the packet departure probability, all algorithms provide similar PL results under either a light congestion or no congestion situation. The DGRED algorithm evidently drops more packets at the router buffer than the RED, GRED, and AGRED algorithms when the probability of packet arrival is higher than the probability of packet departure. Similarly, the reason for this result is because the router buffer in the DGRED algorithm overflows at an earlier time compared with those in RED, GRED, and AGRED [15].

Generally, the disadvantages of the existing congestion control algorithm can be summarized as follows. With bursty traffic, a heavy congestion signal is given out, which then leads to large packet drops. Conversely, network performance becomes degraded when the probability of packet dropping is set too low. Specifically, Dp, PL, mql, and D will increase, and T will decrease. Consequently, a dynamic mechanism is required to implement packet dropping based on the congestion status. Improving network performance involves alleviating PL and obtaining more satisfactory performance measurement results with reference to mql and D when heavy congestion occurs at the router buffers of networks.

Fig.3 Single router buffer for DGRED

III. COMPARISON BASED ON MQL, THROUGHPUT, DELAY, PACKET LOSS and Dp

The mql for all algorithms is identical up to a certain value of the probability of packet arrival (e.g., 0.33). In such a low probability value, there is at most a light congestion state because the probability of packet arrival is lower than that of packet departure (a < B). In such case, all the compared algorithms sustain a good and stable mql. However, for a higher probability value, congestion is more likely to exist at the router buffers. Accordingly, the mql of the AQM algorithms increases exponentially.

AGRED and DGRED have good performance in terms of delay. This result is due to the fewer dropped packets in DGRED than those in RED, GRED, and AGRED.
### IV. COMPARISON TABLE

<table>
<thead>
<tr>
<th>Mechanism / Parameters</th>
<th>RED</th>
<th>GRED</th>
<th>AGRED</th>
<th>DGRED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AQL</strong></td>
<td>Higher AQL</td>
<td>Lower than RED</td>
<td>Lower than RED</td>
<td>Lower than RED</td>
</tr>
<tr>
<td></td>
<td>Higher than RED</td>
<td>similar to AGRED, DGRED</td>
<td>similar to GRED, DGRED</td>
<td>similar to AGRED, DGRED</td>
</tr>
<tr>
<td></td>
<td>if packet arrival probability is not high</td>
<td>if packet arrival probability is not high</td>
<td>if packet arrival probability is not high</td>
<td>if packet arrival probability is not high</td>
</tr>
<tr>
<td><strong>THROUGHPUT (T)</strong></td>
<td>Throughput is similar to GRED, AGRED, DGRED</td>
<td>Throughput is similar to RED, AGRED, DGRED</td>
<td>Throughput is similar to RED, AGRED, DGRED</td>
<td>Throughput is similar to RED, AGRED, DGRED</td>
</tr>
<tr>
<td></td>
<td>Througput is similar to RED, AGRED, DGRED</td>
<td>Througput is similar to RED, AGRED, DGRED</td>
<td>Througput is similar to RED, AGRED, DGRED</td>
<td>Througput is similar to RED, AGRED, DGRED</td>
</tr>
<tr>
<td><strong>DELAY (D)</strong></td>
<td>Higher delay</td>
<td>Lower than RED</td>
<td>Lower than RED</td>
<td>Lowest</td>
</tr>
<tr>
<td></td>
<td>if packet arrival probability is not high</td>
<td>similar to AGRED, DGRED</td>
<td>similar to GRED, DGRED</td>
<td>than RED and greater than DGRED and AGRED</td>
</tr>
<tr>
<td><strong>PACKET LOSS (PL)</strong></td>
<td>Higher packet loss</td>
<td>Lower than RED and greater than DGRED and AGRED</td>
<td>Lower than RED and GRE D and higher than DGRED</td>
<td>Lowest</td>
</tr>
<tr>
<td></td>
<td>Higher than RED and GRE D and AGRED</td>
<td>Packet drop is lower than RED</td>
<td>Packet drop is lower than RED</td>
<td>Packet drop is lower than RED</td>
</tr>
<tr>
<td><strong>PACKAGE DROP (DP)</strong></td>
<td>Packet drop is low</td>
<td>Packet drop is lower than RED</td>
<td>Packet drop is lower than RED</td>
<td>Highest</td>
</tr>
<tr>
<td></td>
<td>Packet drop is lower than RED</td>
<td>Packet drop is lower than RED</td>
<td>Packet drop is lower than RED</td>
<td>Packet drop is lower than RED</td>
</tr>
</tbody>
</table>

### V. CONCLUSION

The RED, GRED, AGRED, and DGRED algorithms provide similar performance measure results (mql, T, D, PL, and Dp) when the probability of packet arrival is set to a value lower than the probability of packet departure or in the event of light or no congestion. AGRED and DGRED have good performance in terms of delay. This result is due to the fewer dropped packets in DGRED than those in RED, GRED, and AGRED. In addition, the RED, GRED, AGRED, and DGRED algorithms obtain similar T results with such values of packet arrival probability.

### REFERENCES