FLoadAutoRED: An Active Queue Management Scheme to Prevent Congestion in a Dynamically Varying Traffic in IP Networks

CHAPTER 4

METHODOLOGY – PHASE II

4.1 PARAMETER SENSITIVITY

AutoRED AQM has parameters that determine the lower, upper bound on the average queue size and the maximum rate for dropping packets. The congestion prevention mechanism should have low parameter sensitivity, and the parameters should be applicable to networks with widely varying bandwidths or varying flows. The AutoRED AQM parameters max_p, min_{th}, and max_{th} are necessary so that the network designer can make conscious decisions about the desired average queue size, the duration in queue and bursts to be allowed at the gateway. The parameter max_p can be chosen from a fairly wide range, because it is only an upper bound on the actual dropping probability p_b. If congestion is sufficiently heavy that the AQM cannot control the average queue size by dropping atmost a fraction max_p of the packets, then the average queue size will exceed the maximum threshold and the AQM will drop every packet at gateway until congestion is controlled.

AutoREDwithARED strives to address this problem of RED’s sensitivity to parameters. RED’s well-known characteristic of the average queue size and performance varies as a function of the RED parameters max_p and w_q. ARED shows that adapting to max_p keeps the target queue size within a target range between min_{th} and max_{th}.

ARED adapts max_{th} in response to measured queue lengths. ARED adjusts max_{th} in response to current conditions, and is effective in achieving high throughput along with maintaining its average queue size within the target interval.

The parameter max_p is adapted not just to keep the average queue size between min_{th} and max_{th} but also to keep the average queue size within a target range half way between min_{th} and max_{th}. The parameter max_p is adapted slowly, and over time its scales greater than a typical round-trip time in small steps. Instead of multiplicatively increasing and decreasing max_p, an additive-increase multiplicative decrease (AIMD) policy is used. The

AutoREDwithRED performs better than AutoREDwithARED in terms of throughput and packet drop rate.

A dramatic oscillation in the average queue size takes place with the average queue size going below min_{th} and above max_{th} in each oscillation. This leads to oscillations between periods of high packet drop rates and periods of no packet drops, and results in degraded throughput and high variance in queuing delay. Exceeding max_{th} incurs a non-linearity in the form of a large packet drop, with a corresponding decrease in utilization, and forces the average queue size to decrease sharply in this case, below min_{th}. When the average queue size falls below min_{th} the average packet drop probability becomes zero. Therefore the importance of queue threshold is discussed in the following section.

4.2 IMPORTANCE OF QUEUE THRESHOLD

To give an adequate performance under a wide range of traffic conditions, AQM parameters play a vital role in AutoRED. Firstly it ensures adequate calculation of the average queue size. The average queue size at the gateway is limited by max_{th}, as long as the calculated average queue size is a fairly accurate reflection of the actual average queue size. The weight w_q parameter is set as dynamic so that the calculated average queue length does not delay too long in reflecting increases in the actual queue length. To maximize network power, the min_{th} is to be set sufficiently high. The thresholds min_{th} and max_{th} should be set sufficiently high to maximize network power. If the typical traffic is fairly bursty, then min_{th} must be correspondingly large to allow the link utilization to be maintained at an acceptably high level.

It is important to determine the optimal average queue size for various network conditions. Because network traffic is often bursty, the actual queue size can also be quite bursty; if the average queue size is kept too low, then the output link will be underutilized. Incase of high average queue size, the output link will give a higher queuing delay. When $max_{th} – min_{th}$ is larger than the typical increase in average queue size, it avoids the global synchronization. Therefore to avoid global synchronization, have max_{th} – min_{th} sufficiently large. If max_{th} – min_{th} is too small then the computed average queue size can regularly oscillate up to max_{th}. Queue threshold plays a vital role in maintaining the queue size. Therefore, in the methodology, the maximum queue threshold max_{th} is made dynamic and updated based on the current queue status and congestion status as the second phase.

4.3 FDynamicAutoRED ALGORITHM

The queue threshold max_{th} of the queue size is an important parameter to decide the packet dropping probability. Therefore this queue threshold indicates the actual queue size maintained by the AQM. To overcome this problem, max_{th} of the queue size should not be a user dependent parameter to very large extent, as a poor user parameter value results in a very lower or higher packet drop probability. Therefore the packet drop probability keeps the average queue size away from the actual queue size and the AQM fails in incipient congestion indication. For this reason the queue size and congestion are not well controlled resulting in a poor performance. So to overcome this problem of poor tuning of the parameter queue threshold max_{th}, it is made more dynamic and user independent. To have a better tuning of the queue threshold max_{th}, the level of congestion is chosen.

To keep the max_{th} dynamic, the level of congestion is to be manipulated. The level of congestion is considered at three stages i.e low congestion, initial stages of heavy congestion and heavy congestion. Based on the level of congestion, the maximum threshold max_{th} is either updated or not. Incase of low congestion or very high congestion, maximum queue threshold max_{th} is not updated. Only incase of initial stages of heavy congestion, the queue threshold is updated.

Therefore in the algorithm shown in Table 4.1, max_{th} of the queue size is not kept constant and it varies depending on the congestion in the network.

Based on the level of congestion in a network at a particular instant of time, max_{th} of the queue is either incremented or decremented. As the probability of packet drop depends on max_{th}, the value of max_{th} is kept dynamically varying based on the level of congestion. The point at which congestion exists is based on certain factors like probability of congestion, instantaneous queue size, initial maximum queue threshold and current maximum queue threshold.

Table 4.1 Pseudocode of FDynamicAutoRED

```
Initially $I_{maxth} = Cur_{maxth} = max_{th}$
For every packet arrival {

Calculate $p_t$
Calculate $w_q$

If ($p_t < 0.550$) && ($Q_t < I_{maxth}$) && ($Cur_{maxth} > I_{maxth}$)
        Reinitialise $Cur_{maxth}$ to $I_{maxth}$
Else if ($p_t > 0.550$) && ($p_t < 0.880$) && ($Q_t >= I_{maxth}$) && ($Cur_{maxth} == I_{maxth}$)
             Increment $Cur_{maxth}$

Calculate $Q_{avg}$

if ($Q_{avg} < min_{th}$)
      Forward the new packet
Else
      Select randomly a packet from the queue for their flow id
      Compare arriving packet with a randomly selected packet.
      if they have the same flow id
             Drop both the packets
      Else
             if ($Q_{avg} >= Cur_{maxth}$)
                  Drop the new packet
             Else
                  Calculate $p_b = max_p . (Q_{avg} - min_{th}) / (Cur_{maxth} – min_{th})$
                  Calculate $p_a = p_b / (1- count . p_b)$
                  Drop the arriving packet with probability $p_a$
}
```

The level of congestion varies and is indicated depending on the following criteria:

- p_t (Probability of congestion)
- Q_t (Instantaneous queue size) and I_{maxth} (Initialmaxthresh)
- Cur_{maxth} (Currentmaxthresh) and I_{maxth} Initialmaxthresh)

As discussed in Chapter 3, the probability p_t that the network can lead to congestion at t+1 based on the queue status at time t is calculated using the following equation:

$$p_t = \frac{n_t}{n_t + n_t^{'}}$$

where n_t and $n_t^{'}$ stand for the number of times $q_i \geq q_{avg,i-1}$ and $q_i < q_{avg,i-1}$ respectively within the time duration t. Hence the network can lead to no congestion or congestion at a particular instant of time and helps to decide whether to update the maximum queue threshold max_{th} or not.

Another factor that is considered to identify the level of congestion is the instantaneous queue size. It also helps to decide the status of the network in terms of congestion. The instantaneous queue size gives clear information about the queue state at a particular instant of time. This queue state helps in deciding whether the network will lead to congestion in future.

The maximum queue threshold is also updated based on the initial maximum threshold set by the user. Cur_{maxth} also helps in taking decision of updating the max_{th} as the queue threshold cannot be kept updated throughout the existence of congestion. Hence the third factor used is the current maximum threshold.

If $(p_t < 0.55)$ && $(Q_t < I_{max_{th}})$ && $(Cur_{max_{th}} > I_{max_{th}})$, then reinitialise Cur_{maxth} to I_{maxth}. This indicates a low congestion or no congestion in the network so Cur_{maxth} can be reinitialized. In case when the probability (p_t) that the network can lead to congestion at time t + 1 is between 0 and 0.55 and the instantaneous queue size is lesser than the initial maximum threshold it is an indication that the queue size is in the safer region and queue size will not increase above max_{th}. Hence the value of max_{th} will not affect the probability of packet drop in case of no congestion or low congestion. Therefore the reinitializing of max_{th} is required if max_{th} has been previously incremented.

If $(0.55 \leq p_t \leq 0.88)\ \&\&\ (Q_t \geq I_{max_{th}})\ \&\&\ (Cur_{max_{th}} = I_{max_{th}})$, then Increment Cur_{maxth}. This indicates initial stages of heavy congestion in the network so Cur_{maxth} is incremented to control the probability of packet drop. The probability (p_t) that the network can lead to congestion at time t + 1 is higher in the range between 0.55 and 0.88. A higher value of max_{th} in the initial stages of heavy congestion will result in reduced probability of packet drop.

Incase of very heavy congestion (i.e. p_t greater than 0.88) which shows the existence of very heavy congestion, then max_{th} will not affect the probability of packet drop because any value of max_{th} will lead to packet drop. The value of max_{th} will be vital only in case of initial stages of heavy congestion and not during no congestion or very heavy congestion. The packet drop probability will be controlled based on the value of max_{th}. In the proposed algorithm, max_{th} is set based on the probability of congestion at time t in network and the occurrence of congestion at time t-1.

So calculating average queue size and packet drop probability are not dependent on well tuning of the parameters as in AutoRED. The proposed algorithm works fine as the parameters are well tuned automatically. In such a case, the proposed algorithm reduces the problem of parameter tuning.

The value of max_{th} decides the packet drop. The threshold value of max_{th} should be set sufficiently to increase the network power. In case of low threshold value of max_{th}, the average queue size will remain too low. If the average queue size is too low, then the output link will be underutilized. When the network traffic is bursty then average queue size can also be made bursty to improve link utilization. Nextly increasing $\mathrm{max}_{\mathrm{th}} - \mathrm{min}_{\mathrm{th}}$ sufficiently large avoids global synchronization. If the $\mathrm{max}_{\mathrm{th}} - \mathrm{min}_{\mathrm{th}}$ is too small, then the computed average queue size will be regularly around $\mathrm{max}_{\mathrm{th}}$. The dynamic varying nature of the parameters w_{q}, $\mathrm{max}_{\mathrm{th}}$ and the flow information will try to keep the router congestion controlled with reduced packet drop rate.

The maximum queue threshold $\mathrm{max}_{\mathrm{th}}$ is updated dynamically as follows:

$$Cur_{\max_{th}} = \left\{ \begin{array}{l} I_{\max th}, \\ \qquad\qquad if\,(P_t < 0.55)\,\&\,\&(Q_t < I_{\max_{th}} < Cur_{\max_{th}}) \\ Cur_{\max_{th}} + 1, \\ \qquad\quad if\,(0.55 < P_t < 0.88)\,\&\,\&(Q_t \geq I_{\max th} = Cur_{\max_{th}}) \end{array} \right\}$$

In Table 4.1, initially the current maximum threshold ($\mathrm{Cur}_{\mathrm{maxth}}$) and initial maximum threshold ($\mathrm{I}_{\mathrm{maxth}}$) one set as the user defined maximum queue threshold $\mathrm{max}_{\mathrm{th}}$. The probability p_t that the network can lead to congestion is calculated based on the queue status. The dynamic value of weighting constant w_{q} is calculated. Based on p_t, instantaneous queue size, initial maximum threshold and current maxthreshold, the current maxthreshold ($\mathrm{Cur}_{\mathrm{maxth}}$) is updated.

The average queue size $\mathrm{Q}_{\mathrm{avg}}$ is then calculated. Based on the minimum threshold and maximum threshold, the packet drop decision is made.

4.4 EXPERIMENTATION

The packet-simulator ns-2 is used to simulate the FDynamicAutoRED algorithm. In this simulation the network topology in Figure 4.1 is with a single link of capacity 1Mbps that drops packet according to the AQM algorithm with the parameter values as in Table 4.2. The congestion link is in between the two routers R1 and R2. In the simulation setup 32 TCP flows and 1 UDP flow is considered in the network.

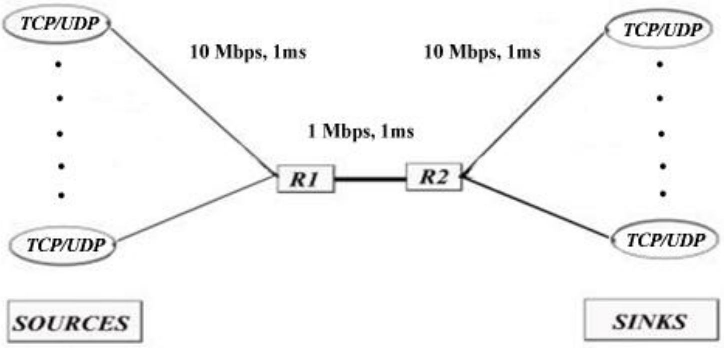

Figure 4.1 Simple Network Topology

Table 4.2 Parameter Setting of FDynamicAutoRED

Type of Sources	**Buffer Size**	**Link Capacity**	**Link Delay**	**Packet Size**	**Minimum Threshold min_{th}**	**Maximum Threshold max_{th}**	**max_p**
TCP (FTP) UDP (CBR)	300 Packets	1Mbps	1ms	1 Kbytes	100 Packets	200 Packets	0.02

In a dynamic varying mixture of traffic, the control parameter w_q alone does not help in achieving the stable operating point for the queue size. As shown in Table 4.3 dynamic varying parameter w_q maintains the average queue size. FDynamicAutoRED algorithm shows a lower average queue size compared to AutoREDwithRED as in Table 4.3. The average queue size is neither too low nor high in this proposed AQM as compared to AutoREDwithRED AQM. A very high average queue size increases the queuing delay. The average queue size should not be too low which results in poor link utilization. The standard deviation reflects the queue stability which is small and tolerable for the AQMs.

The queue threshold max_{th} reflects the change in average queue size and the packet drop rate. The dynamic queue threshold increases the queue threshold resulting in a better utilization. The dynamic parameter queue threshold shows a reduced packet dropped rate. The proposed algorithm FDynamicAutoRED keeps the packet drop rate minimised compared to AutoREDwithRED. The results are shown in Table 4.4.

Table 4.3 Comparison of Average Queue Size / Std. Dev of AQMs

AQMs	Average Queue Size (# Packets)	Std. Dev. Avg Queue Size
AutoREDwithRED	190.4	3.5
FDynamicAutoRED	**100.8**	**5.2**

Table 4.4 Comparison of Packet Drop Rate of AQMs

AQMs	Packet Drop Rate (%)	FDynamicAutoREDwith RED Improvement over Other AQMs (%)
AutoREDwithRED	9.74	0.25
FDynamicAutoRED	**9.49**	**-**

AutoREDwithRED is unable to penalize unresponsive flows, as the packets dropped from each flow over a period of time is almost the same. Consequently the misbehaving traffic like UDP can take up a large percentage of the link bandwidth and starve out TCP friendly flows. The results are shown in Table 4.5**.** FDynamicAutoRED identifies and penalizes misbehaving flows effectively compared to the existing AQM as in Table 4.5.

Table 4.5 Comparison of Utilisation of AQMs

AQMs	**In %**		
	CBRutilisation	**TCPutilisation**	**OverallUtilisation**
AutoREDwithRED	97.29	1.70	99.00
FDynamicAutoRED	**21.25**	**77.18**	**98.44**

It is obvious from the experimentation that the existing AQM almost takes up the entire bandwidth for UDP flow though its actual CBR fair share is very minimum. The CBR throughput is only 21% of the bandwidth using this algorithm. So TCP utilization is almost very minimum and this algorithm shows a good fair utilization of 77%.

The FDynamicAutoRED has a controlled packet drop compared to the existing AQM due to the dynamic queue threshold. It tries to give a fair share by dropping the UDP packets otherwise the UDP utilises the link to the maximum without allowing the TCP packets. The packet drop rate is controlled due to the increment/decrement in the max_{th} that depends on the congestion characteristics.

The dropping probability is adjusted such that the packets of the flow with higher transmission rate are more likely dropped than flows with lower rate. The fairness obtained by this AQM is better for the TCP flows as shown in Table 4.6. The good performance of proposed AQM in disciplining TCP flows is observed in the results. However, in presence of overwhelming, unresponsive UDP traffic, the other AQM is less effective in achieving good fairness.

Table 4.6 Comparison of Fairness of AQMs

AQM	Fairness	FDynamicAutoRED withRED Improvement over Other AQMs (%)	Variation	FDynamicAutoRED withRED Improvement over Other AQMs (%)
AutoREDwithRED	0.5	16.6	0.22	95.45
FDynamicAutoRED	**0.61**	**-**	**0.01**	**-**

A simple modification to the FAutoREDwtihRED algorithm makes it capable of handling unresponsive flows with the reduced packet loss. During the packet arrival rate at queue, only the fair share throughput from the source is admitted to the queue, and the rest of the packets are dropped.

4.5 CONCLUSION

The proposed methodology removes the parameter tuning problem by implementing the dynamic queue threshold max_{th}. This results in avoiding global synchronization. The AQM scheme FDynamicAutoRED aims to protect the adaptive flows from non-adaptive flows. It is obtained without comprising high utilisation, low queuing delay and controlled packet loss. It is achieved with dynamically tuned parameters and flow information. The packet dropping scheme discriminates against the unresponsive flows resulting in fair congestion indication. It obtains a higher TCP utilization and also overall utilization.

The vulnerability of FDynamicAutoRED has been experienced in disciplining misbehaving flows and achieving adjustable fairness in a varying traffic in IP network where there is a need to prevent unresponsive flows from overwhelming others. Additionally the performance of the varying traffic is improved by reducing the probability of experiencing packet losses for such traffic. It tries to control packet drop rate but it does not still attain a moderate average queue size. A queue based AQM needs to improve its adaptability i.e responsive performance compared to a load based AQM. Therefore an improvement is required in terms of adaptability for this queue based AQM. The next section discusses the proposed method for adaptability.

Publication

- **International Journal**

i. "FDynamicAutoRED: An Algorithm to Stabilise the Queue in Internet Routers", International Journal of Computer Applications, ISSN 0975 – 8887, ISBN 978-93-80749-28-0, Vol. 21 No.7, pp. 20 – 25, May 2011. Impact Factor - 0.835.