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

CHAPTER 6

METHODOLOGY – PHASE IV

6.1 IMPORTANCE OF UTILISATION AND QUEUING DELAY

For AQM, the performance objectives include efficient queue utilization with regulated queuing delay and robustness.

i. Efficient *queue utilization:* For efficient use, the queue should avoid overflow or emptiness. The former situation results in lose of packets and undesired retransmissions, while an empty buffer under-utilizes the link. Both of these extremes should be avoided in both transient and steady-state operations.

ii. *Queuing delay:* The time required for a data packet to be serviced by the routing queue is called the queuing delay. Together with the propagation delay t_p, the time accounts for the network delay and it is desirable to keep small both the queuing delay and its variations. This calls for regulating to small queue lengths; however, doing so may result in link underutilization and this limitation presents a fundamental tradeoff to AQM design.

iii. *Robustness:* AQM schemes need to maintain closed-loop performance in face of varying network conditions. These conditions include variations in the number of TCP sessions N.

The first feature implies that, contrary to the conventional wisdom, high utilization is not achieved by keeping large backlogs in the network, but by feeding back the right information for users to set their rates. An AQM should maintain high utilization with negligible loss or queuing delay as the number of user increase.

The second feature is essential in a network where users typically go through congested links. It clarifies the meaning of the congestion information embedded in the dropping probability observed by a user, and thus can be used to design its rate adaptation.

High link utilization may result in large queues and long delays. The instantaneous utilization is unity when a large queue backlog exists at the bottleneck link. The goal is to maximize the link utilization while keeping the steady-state queuing delay in the neighborhood of desired value. The trade-off between maximizing throughput (utilization) and minimizing delay is the main topic of Internet congestion prevention. An AQM can achieve the goal by maintaining a packet dropping probability corresponding to the link utilization.

6.2 IMPORTANCE OF PACKET DROP PROBABILITY

In light of the above discussions, AQMs also maintain a single probability, which it uses to drop packets when they are queued. If the queue is continually dropping packets due to buffer overflow, AQM increments the dropping probability, thus increasing the rate at which it sends back congestion notification. Conversely, if the queue becomes empty or if the link is idle, AQM decreases its dropping probability. Link underutilization can occur when congestion management is either too conservative or too aggressive, but packet loss occurs only when congestion management is too conservative.

The control aim of the queue-based AQM schemes is to stabilize the queue to desired value. Queue length is the cumulative of the difference between arrival rate and output rate at the links. However, queue based AQM schemes make slow action to congestion. At the same time, congestion dropping behavior is conservative when the queue size is small but increases rapidly. However, congestion dropping behavior is aggressive when the queue length is large but decreases quickly. Although the rate based AQM schemes can obtain good transient performance because of its fast responsiveness, small queuing may occur by reason of no explicit control mechanism of queue size under dynamic network scenarios especially.

AutoRED interacts with TCP as source rates increase, queue length grows, more packets are dropped, prompting the sources to reduce their rates, and the cycle repeats. TCP defines precisely how the source rates are adjusted while active queue management defines how the congestion measure is updated. For AutoRED, the congestion measure is queue length and it is automatically updated by the buffer process.

AutoRED measures congestion with average queue length, but the critical point is that it couples congestion measure with performance measure. It is thus inevitable that the average queue under AutoRED grows with the number of users. With the AutoRED, it can grow to the maximum queue threshold $\max_{th}$ where all packets are dropped. If $\max_{th}$ is set too high, the queueing delay can be excessive; if it is set too low, the link can be underutilized due to severe buffer oscillation. Moreover, if congestion signal is fed back through random dropping, packet losses can be very frequent. Hence in times of congestion, AutoRED can be either tuned to achieve high link utilization or low delay and loss, but not both. In contrast, by updating the packet dropping probability, queue can be stabilized around its target independent of traffic load, leading to high utilization and low delay.

If the AutoRED AQM in fact drops packets arriving at the gateway when the average queue size reaches the maximum threshold, then the AutoRED gateway guarantees that the calculated average queue size does not exceed the maximum threshold.

In AutoRED, as average queue size Q_{avg} varies from $\min_{th}$ to $\max_{th}$, the packet-dropping probability p_b varies linearly from 0 to $\max_p$, final packet-dropping probability p_a increases slowly as the count increases since the last dropped packet.

$$p_b = \frac{Q_{avg} - \min_{th}}{\max_{th} - \min_{th}} \cdot \max_p$$

$$p_a = \frac{p_b}{1 - count \cdot p_b}$$

Some AQMs maintain a single probability that corresponds to the link utilization or link idle to drop packets when they are enqueued. An AQM also maintains dropping probability corresponds to the link utilization. If current link utilization is less than the desired link utilisation, packet dropping probability decreases, otherwise, it increases. The key idea behind such AQMs is to perform queue management based directly on link load factor. This originates from the primary aim for congestion control, which is to achieve high link utilization with low queuing delay and to avoid high packet loss rate. Therefore, such AQMs responds fast when there exists a rate mismatch between input rate and link capacity. An AQM is discussed in the next section with the packet drop probability corresponding to the link utilization.

6.3 METHODOLOGY - FLoadAutoRED ALGORITHM

The pseudocode of FLoadAutoRED is in Table 6.1. The objective of FLoadAutoRED is to maximize the link utilization with low queuing delay. This is made possible by allocating as much of the available capacity to all active flows in the network. The proposed AQM tracks link utilization by using the load factor ‘z’. Using ‘z’, the AQM tries to achieve the steady-state operating point in the neighborhood of $z = 1$. Then, the high link utilization can be achieved by controlling the load factor ‘z’ in the domain $[1, 1+\delta]$. Hence the parameter ‘δ’ defines the steady-state operating region, towards which the AQM attempts to drive the network. When the system is in the steady-state region $[1, 1+\delta]$, the packet dropping probability would be unchanged, then the resource can be allocated fairly with probability 1 by using the stochastic dropping in router. Therefore a suitable value for ‘δ’ is important for steady performance and utilization.

The update value of dropping probability is based on the queue threshold and it also corresponds to the desired link utilization. The desired link utilization is in the interval $[1, 1+\delta]$. The load factor ‘z’ is considered as the current link utilization. If current link utilization ‘z’ is less than desired utilization then packet dropping probability decreases, otherwise, it increases.

Table 6.1 Pseudocode of FLoadAutoRED

```
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 $Q_{avg}$

VQC = $\gamma\,(\lambda(t) - C)$
z = $\lambda(t)$ / VQC

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 the dropping probability $p_a$
                Calculate $p_b$= $max_p$ . ($Q_{avg}$ - $min_{th}$) / ($Cur_{maxth}$ – $min_{th}$)
                Calculate $p_a$= $p_b$ / (1- count . $p_b$)

                if (z >= 1 + $\delta$ )
                  Calculate P = $p_a$ + (z*Delta)/C
                else
                if  ( z < 1)
                  Calculate P = $p_a$ – (Delta/zC)
                else
                  Calculate P = $p_a$

                Drop the packet with probability P

}
```

In the first stage, the packet drop probability is calculated based on the queue size. Based on the average queue size, the packet drop probability varies and is calculated as follows:

$$p_b = \frac{Q_{avg} - \min_{th}}{\max_{th} - \min_{th}} \cdot \max{}_p$$

$$p_a = \frac{p_b}{1 - count \cdot p_b}$$

In the second stage, based on the load factor the packet drop probability is updated. The packet drop probability P is updated as follows:

$$P = \begin{cases} p_a + (z \cdot Delta)/C, & \\ & if\ \ (z \geq 1 + \delta) \\ p_a - (Delta/z \cdot C), & \\ & if\ \ (z < 1) \\ p_a\,, & Otherwise \end{cases}$$

where

P	Current Packet Drop Probability
p_a	Packet drop probability based on Queue
z	load factor
Delta	Reflects the value of response ability for the congestion
C	Link Capacity
δ	Stable Interval

The value of Delta reflects response ability for the congestion signal. Larger value makes the speed of response quickly with the possibility of larger queue oscillations, while smaller value limits the speed of response to steady state. The performance influence of parameter ‘δ’ is in terms of stability. The larger the stable interval ‘δ’, the larger is the average queue length.

6.4 EXPERIMENTATION

The algorithm FLoadAutoRED is simulated. The simulation parameters are shown in Table 6.2. Figure 6.1 shows the network topology. The congestion link is in between the two routers R1 and R2. The UDP hosts send packets at a constant bit rate of 2 Mbps. In the simulation setup 32 TCP flows and 1 UDP flow are considered for the simple network.

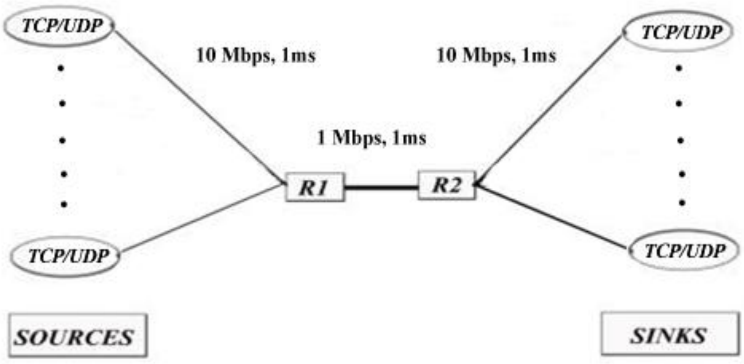

Figure 6.1 Simple Network Topology

Table 6.2 Parameter Setting of FLoadAutoRED

Type of Sources	**Buffer Size**	**Packet Size**	**min$_{th}$**	**max$_{th}$**	**max$_p$**	γ	**Delta**	δ
TCP (FTP) UDP (CBR)	300 Packets	1 Kbytes	100 Packets	200 Packets	0.02	0.2	11.25	0.06

In a dynamic varying mixture of traffic, as discussed in previous chapters, the control parameter w_q alone does not help in achieving the stable operating point for the queue size. AutoREDwithRED keeps the average queue size at a stable point for the traffic but keeps the queue size at higher level to show performance degradation. On the other hand, FLoadAutoRED shows a moderate average queue size compared to other AQM as in Table 6.3**.** However, a small increase in the standard deviation of queue size does not affect the performance of this AQM. The average queue size is neither too low nor high and keeps the average queue size controlled at a moderate level.

FLoadAutoRED also achieves better adaptability by adapting the virtual queue capacity to the packet arrival rate. The packet drop probability is decreased or increased based on the adaptability required in terms of load factor ‘z’. Therefore the proposed AQM shows a reduced packet drop rate as in Table 6.4. It shows an improvement over AutoREDwithRED. FLoadAutoRED also shows higher link utilization as in Table 6.5 and irrespective of higher link utilization it shows a lower queuing delay. The average queue size is maintained at a moderate level to keep the queue delay low compared to AutoREDwithRED.

Table 6.3 Comparison of Average Queue Size / Std. Dev of AQMs

AQMs	Average Queue Size (# packets)	Std Dev. Avg. Queue Size (# packets)
AutoREDwithRED	190.4	3.5
FLoadAutoRED	**145.6**	**6.3**

Table 6.4 Comparison of Packet Drop Rate of AQMs

AQMs	Packet Drop Rate (%)	Improvement of FLoadAutoRED over Other AQMs (%)
AutoREDwithRED	9.74	4.38
FLoadAutoRED	**5.39**	**--**

Table 6.5 Comparison of Utilisation/Queuing Delay of AQMs

AQMs	Queuing Delay (ms)	Improvement of FLoadAutoRED over Other AQMs (%)	Overall Utilisation (%)	Improvement of FLoadAutoRED over Other AQMs (%)
AutoREDwithRED	0.18	33.3	99.1	0.6
FLoadAutoRED	**0.12**	**--**	**99.77**	**--**

The throughput of the entire TCP is affected by UDP traffic. For all AQM schemes, the TCP throughput is reduced when UDP traffic increases as shown in Table 6.6. In spite of the existence of UDP flows, FLoadAutoRED outperforms the other schemes for link utilization of TCP traffic and the fairness as in Table 6.7. The fairness and variation achieved by the proposed algorithm is 0.63 and 0.01 respectively.

Table 6.6 Comparison of TCP/CBR Utilisation of AQMs

AQMs	In %		
	CBRutilisation	TCPutilisation	OverallUtilisation
AutoREDwithRED	97.29	1.70	99.00
FLoadAutoRED	**18.25**	**81.52**	**99.77**

Table 6.7 Comparison of Fairness of AQMs

AQMs	Fairness	Improvement of FLoadAutoRED over Other AQMs (%)	Variation	Improvement of FLoadAutoRED over Other AQMs (%)
AutoREDwithRED	0.5	16.6	0.22	95.45
FLoadAutoRED	**0.63**	**--**	**0.01**	**--**

6.5 CONCLUSION

The inherent weakness of current active queue management algorithms is demonstrated. The key design goal of FLoadAutoRED is to predict incipient congestion timely and accurately with controlled queuing delays, stability, and robustness to variation in traffic load. In proposed scheme, the routers periodically monitor their load on each link and determine a load factor, the available capacity and the queue length. This information is used to alert the incipient congestion in advance and the packet dropping probability is calculated. The proposed scheme meets the desired goals, including good steady-state behavior with high utilization, controlled queuing delay, fairness and robustness to traffic variations.

Publication

- **International Journal**

i. "FLoadAutoRED: An AQM scheme to Increase the Overall Performance in Internet Routers", International Journal of Computer Science Issues, Volume 8, Issue 4, pp. 308 – 312, July 2011. Impact Factor – 0.242