

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

CHAPTER 5

METHODOLOGY – PHASE III

5.1 IMPORTANCE OF LOAD BASED AQMs

5.2 VIRTUAL BASED AQMs

5.3 IMPLEMENTATION OF LOAD AND VIRTUAL QUEUE – METHODOLOGY

5.4 EXPERIMENTATION

5.5 CONCLUSION

5.1 IMPORTANCE OF LOAD BASED AQMs

One of the fundamental problems with AutoRED technique is that they rely only on queue length as an estimator of congestion. Though the presence of a persistent queue indicates congestion, its length gives very little information as to the severity of congestion, that is, the number of competing connections sharing the link. During the congestion period, even when a single source transmits at a rate greater than the bottleneck link capacity can cause a queue to build up just as easily as a large number of sources do. From well-known results in queuing theory, it is only when packet interarrivals have a Poisson distribution that queue lengths directly relate to the number of active sources and thus the true level of congestion. Unfortunately, packet interarrival times across network links are decidedly non-Poisson. Packet interarrivals from individual sources are driven by TCP dynamics and source interarrivals themselves are heavy-tailed in nature. Therefore a queue-based active queue management schemes become dubious. Since the queue-based algorithm relies on queue lengths, it has an inherent problem in determining the severity of congestion. As a result, queue-based algorithm requires a wide range of parameters to operate correctly under different congestion scenarios.

In queue-based AQMs, when the queue is fully occupied, AQMs overmark and drop packets causing a subsequent period of underutilization. Conversely, when the queue is empty, AQM undermarks packets causing a subsequent period of high packets loss as the offered load increases well beyond the link's capacity. AQMs sustain increasingly high packet loss as the number of connections is increased. Since aggregate TCP traffic becomes more aggressive as the number of connections increase and it becomes difficult for AQMs to maintain low loss rates. Fluctuations in queue lengths occur so abruptly that the AQMs algorithm oscillates between periods of sustained dropping and packet loss to periods of minimal dropping and link utilization.

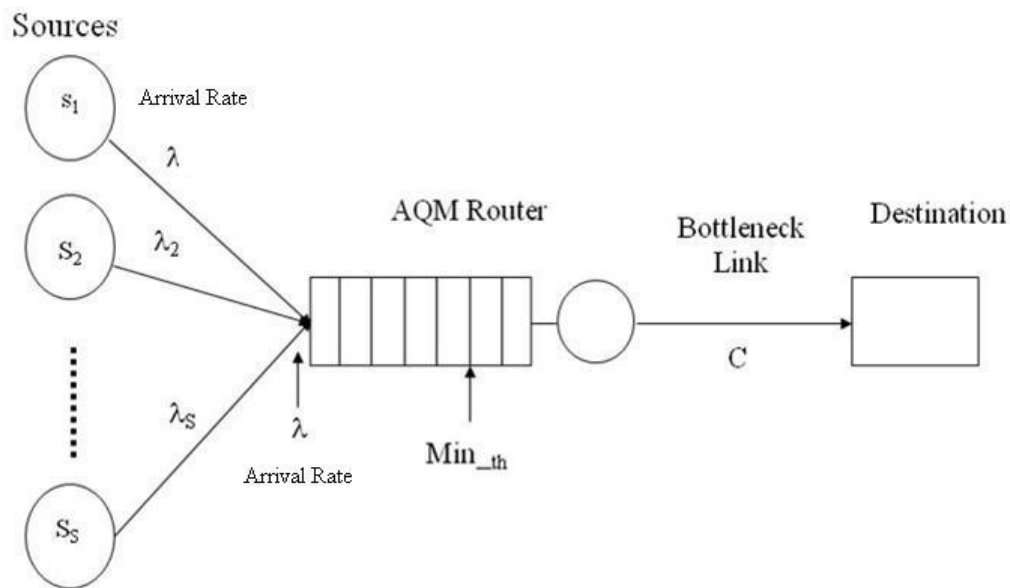


Figure 5.1 Simple Network Model with AQM

Consider a network with flows whose packet sending rate changes with time as in Figure 5.1. Let $\lambda(t)$ be the arrival rate of such a flow. This change of rate of the source may introduce jitter in the routers. It is assumed that ‘ μ ’ is the service rate of the router and ‘ n ’ is the number of active flows.

Consider the overload factor U as follows:

$$U = \frac{\lambda}{\mu}$$

If the overload-factor U is below 1, then router is non-congested and packets are not marked or dropped. Therefore the low congestion ($\lambda \approx \mu$), does not effect the queue. Further, in case of high congestion ($\lambda \gg \mu$), overload-factor U is above 1, router is congested and it affects the queue and more packets are dropped at the queue. Hence, DropRatio(p) at the router during high congestion is

$$p = 1 - \frac{\mu}{n\lambda}$$

For any change in arrival rate from λ to λ' , buffer (instantaneous length q) will start building up at the rate of

$$\left[\frac{dq}{dt} \right]_{\lambda=\lambda'} = \lambda' (1-p) - \frac{\mu}{n}$$

For small change in λ , $\lambda' = \lambda + d\lambda$, change in queue building rate will be

$$\frac{d}{d\lambda} \left(\frac{dq}{dt} \right) = \left[\frac{dq}{dt} \right]_{\lambda=\lambda+d\lambda} - \left[\frac{dq}{dt} \right]_{\lambda=\lambda'}$$

as 'p' remain same during that period.

$$\frac{d^2q}{dt d\lambda} = \frac{\mu}{n\lambda}$$

Thus $\frac{dq}{dt}$ can be written in terms of λ as,

$$\frac{dq}{dt} = \frac{\mu}{n} \log \lambda - \frac{\mu}{n} \log \lambda_0$$

$$\frac{dq}{dt} = \frac{\mu}{n} \log \left(\frac{\lambda}{\lambda_0} \right)$$

where λ_0 is the value of λ at time $t=0$. This result can be generalized by considering aggregate arrival rate. For aggregate arrival rate λ ,

$$\frac{dq}{dt} = k \log(\lambda), \text{ where } k = \frac{\mu}{n} \text{ is a constant}$$

The above discussion implies that with the change in arrival rate at the router, queue builds up at a rate that introduces congestion. The change in arrival rate has implications in queue buildup and jitter properties of the router.

Therefore additionally, queue state also depends on arrival rates. Hence the load state such as the packet arrival rate provides a clue about the change of the queue state and hence can be used to predict the forthcoming state. Using the average queue length and the arrival rate, average queue length can be predicted to achieve better adaptability by maintaining a virtual queue.

5.2 VIRTUAL QUEUE BASED AQMS

Most of the AQM schemes involve adapting the dropping probability irrespective of whether it is a queue-based, load based or both. Thus an important question arises to how fast should one adapt while maintaining the stability of the system? The system refers jointly to the TCP congestion controllers operating at the edges of the network and the AQM schemes operating in the interior of the network. Adapting too fast might make the system respond well to changing network conditions, but it might lead to large oscillatory behavior or in the worst case even instability. Adapting it too slowly might lead to sluggish behavior and more losses or marks than desired which might lead to a lower throughput.

The adaptability can be stabilized by maintaining a virtual queue. The capacity of the real queue is the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is updated to reflect a new arrival. If the new packet overflows the virtual queue, then the packet is discarded in the virtual queue and the real packet is dropped. Packet in the real queue is dropped at the router. The virtual capacity at each link is then modified such that total flow entering each link achieves a desired utilization of the link.

A virtual queue based AQM scheme either adapts the service rate of the virtual queue or it fixes the service rate as the link capacity of the real queue and adapts the limit of the virtual queue to the packet arrival rate. A packet is dropped whenever the virtual queue overflows the physical buffer limit. It also maintains a virtual queue to adapt the link capacity to the current packet arrival rate to achieve a certain level of link utilization. Therefore this provides an early feedback.

The virtual queue-based scheme addresses the performance, queue size and stability issues more effectively than queue based AQMs. It maintains a virtual queue and responds to the traffic dynamics faster for better stability, especially in the presence of varying flows. In this proposed scheme, it fixes

the service rate as the link capacity of the real queue and adapts the limit of the virtual queue to the packet arrival rate. This is a mechanism for finding the proper rate at which adaptability should take place to stabilize.

5.3 IMPLEMENTATION OF LOAD AND VIRTUAL QUEUE – METHODOLOGY

The main objective of the proposed algorithm is to bring in better adaptability i.e system response to varying arrival rate. The pseudocode is shown in Table 5.1 that implements better adaptability considering the arrival rate. At each packet arrival, the virtual queue capacity is updated, so that the virtual queue capacity is adapted to the arrival rate λ . For instance, λ is the current aggregated packet arrival rate at the real queue and γ is a constant that determines how fast virtual queue capacity adapts to λ .

The virtual queue capacity is updated as follows:

$$VQC = \begin{cases} C, & \text{if } \lambda(\tau) \leq C \\ \gamma(\lambda(t) - C), & \text{if } \lambda(\tau) > C \end{cases}$$

The VQC implies that more packets are dropped if λ exceeds the link capacity C . The load factor z is calculated as follows:

$$z = \begin{cases} >1, & \text{if } \lambda(\tau) > VQC \\ 1, & \text{if } \lambda(\tau) = VQC \\ <1, & \text{if } \lambda(\tau) < VQC \end{cases}$$

The load factor ‘ z ’ indicates the level of difference between the packet arrival rate and the virtual queue capacity. The proposed AQM responds fast when there exists rate mismatch between input rate and virtual capacity. The concept helps to perform better queue management based directly on link load factor. The value of ‘ z ’ varies based on the difference between the two factors. If the arrival rate greater than the virtual queue capacity, then the ‘ z ’ value is greater than 1. If the arrival rate and the capacity matches then the ‘ z ’ value is 1 otherwise the value is less than 1.

Table 5.1 Pseudo code of FLoadAutoRED

<p>Initially $I_{\max th} = Cur_{\max th} = \max_{th}$ For every packet arrival {</p> <p>Calculate p_t Calculate w_q</p> <p>If $(p_t < 0.550) \ \&\& \ (Q_t < I_{\max th}) \ \&\& \ (Cur_{\max th} > I_{\max th})$ Reinitialise $Cur_{\max th}$ to $I_{\max th}$</p> <p>Else if $(p_t > 0.550) \ \&\& \ (p_t < 0.880) \ \&\& \ (Q_t \geq I_{\max th}) \ \&\& \ (Cur_{\max th} == I_{\max th})$ Increment $Cur_{\max th}$</p> <p>Calculate Q_{avg}</p> <p>Calculate $VQC = \gamma (\lambda(t) - C)$ Calculate $z = \lambda(t) / VQC$</p> <p>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 $(Q_{avg} \geq Cur_{\max th})$ Drop the new packet Else Calculate $p_b = \max_p \cdot (Q_{avg} - \min_{th}) / (Cur_{\max th} - \min_{th})$ Calculate $p_a = p_b / (1 - count \cdot p_b)$ Drop the arriving packet with probability p_a</p> <p>}</p>
--

5.4 EXPERIMENTATION

To simulate the FLoadAutoRED algorithm, a single link of capacity 1Mbps as in Figure 5.2 that drops packet according to the AQM algorithm with parameters set as in Table 5.2. The congestion link is in between the two routers R1 and R2. The link is shared by TCP flows and UDP flows. The TCP flows are derived from FTP sessions. The UDP hosts send packets at a constant bit rate of 2 Mbps. In the simulation setup shows 32 TCP flows and 1 UDP flow is considered for the simple network.

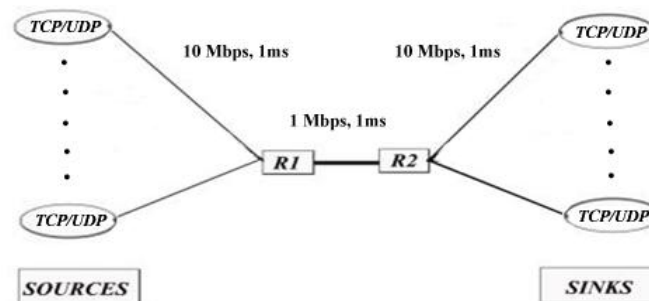


Figure 5.2 Simple Network Topology

Table 5.2 Parameter Setting of FLoadAutoRED

Type of Sources	Link Capacity	Link Delay	Packet Size	\max_p	γ	Delta	δ
TCP (FTP) UDP (CBR)	1Mbps	1ms	1 Kbytes	0.02	0.2	11.25	0.06

In a dynamic varying mixture of traffic, the dynamic parameter w_q and math helps in achieving the stable operating point for the queue size. As shown in Figure 5.3 the queue based AQM shows a poor adaptability compared to the load-based AQM. AutoREDwithRED keeps the average queue size at a high value which is to be improved. However the proposed algorithm shows a stable

and a moderate average queue size compared to other AQMs. The average queue size is neither too low nor high in this proposed AQM as compared to other AQMs. 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 proposed algorithm keeps the average queue size controlled at a moderate value compared to other AQMs.

Figure 5.3 shows the adaptability of the AQM FLoadAutoRED. Adaptability is achieved 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'. The average queue size is maintained at a moderate value to keep the queue delay low. The adaptability improves the system response performance.

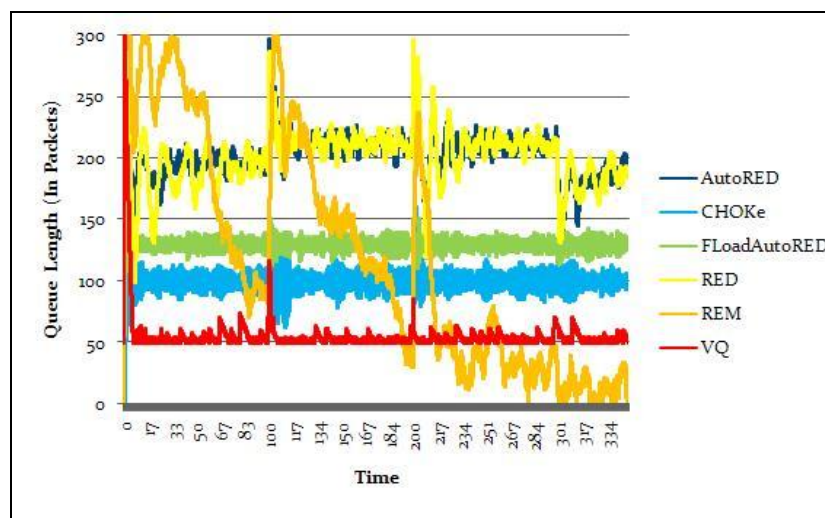


Figure 5.3 Adaptability of AQMs

The proposed AQM scheme uses the packet arrival rate and queue size to decide the rate of adaptability to minimize the effect of network congestion. Comparison with the prior AQM schemes indicates that the proposed algorithm outperforms the other schemes by being more resilient to dynamic workloads in maintaining a stable queue. Stability of the queue is a desirable feature of an

AQM policy since it helps in lowering the packet loss rate. The proposed AQM response fast and maintains a small queuing delay and jitter in a dynamic traffic scenario. The queue based AQM leads to sluggish response and performance degradation in terms of packet loss and queuing delay.

5.5 CONCLUSION

The FLoadAutoRED AQM scheme aims to improve the overall performance of the Internet routers. It protects well-behaved flows from misbehaving flow and adaptive flows from non-adaptive flows. The proposed AQM scheme inherits the advantages of queue length based schemes and input rate based scheme to bring good adaptability of the network. The performance of good service is achieved even under heavy load conditions and protects the responsive flows from unresponsive flows to achieve a good QoS to all users by simulation. The packet drop probability is updated with respect to load factor to show a better system utilisation in the next section.