Using median filtering in active queue management for telecommunication networks

Sorin ZOICAN*, Ph.D.

Cuvinte cheie. Managementul cozilor de așteptare, filtru median, probabilitate de rejectare, întârziere.

Rezumat. Articolul evaluază performanțele unui algoritm RED (random early detection) modificat care utilizează un filtru median în locul unui filtru liniar pentru calculul probabilității de rejectare a pachetelor. Simulările ilustrează faptul că performanța rețelei crește prin utilizarea filtrării neliniare.

Key words. Active queue management, median filter, drop probability, delay

Abstract. This paper evaluates the performance of a modified random early detection (RED) scheme that involves a median filter instead a linear one for drop probability computation. The simulations show that the network performance is better than in the linear filtering case in terms of lower drop probability and delay.

1. Introduction

In the hierarchy of traditionally network design, the signal processing has been considered to the two ends of the protocol stack: the physical layer and the application layer.

Active queue management involves many unsuspected signal processing tasks related to modeling and prediction of congestion, detecting changes in traffic, and even in estimating the number of users or active flows.

A key feature of transport protocol TCP is that it decreases the sending rate when the packet loss appears. The active queue management (AQM) takes advantage of this feature by dropping packets in order to control the packet arrival rates.

The primary objective of AQM is to keep the average queuing delay low while maximizing the throughput. If the queue never empties, then the outgoing link is always busy, hence the highest possible throughput is achieved. While large average queue occupancy reduces the possibility of the queue emptying and the link going idle, it also increases the time it takes for packets to pass through the queue. A balance is desired where the average queue occupancy is as small as possible without causing a significant decrease in throughput.

Another objective of AQM scheme is no bias to bursty traffic. TCP does not send packets at a constant rate; rather it sends packets in bursts with the time between bursts equal to one round trip time (RTT). For a fixed bit-rate, if the RTT is larger, then the burst is larger. If an AQM scheme tends to drop packets that arrive in bursts, then it will tend to drop packets belonging to flows with larger RTT. This would allow flows with shorter RTT to dominate the link bandwidth.

The active queue management schemes reduce the possibility of global synchronization. Suppose
two flows share a link and compete for bandwidth over this link. If these flows also cross some links that are not shared and, consequently, have different round-trip time, it is possible that the flows will become synchronized in such a way that one flow achieves a far higher sending rate than the other flow. Small changes in the link propagation delays often break this synchronization. An effective way to stop synchronization is to drop packets probabilistically. As long as an AQM scheme drops packets randomly, no special design consideration is required to avoid synchronization.

The stability of different AQM algorithms should be considered. Since all quantities are bounded, a stability concept such as bounded input implies bounded output is not applicable. Instead, the local stability should examine. However, if the queue occupancy oscillates but never emptied and never exceeds the specified limit, then these oscillations might be harmless. On the other hand, an AQM that oscillates may lead to unforeseen problems for other protocols that rely on AQM.

Another critical objective of an AQM scheme is simplicity.

2. Description of the AQM Scheme

The random early detection RED algorithm calculates the average queue size, using a low-pass filter with an exponential weighted moving average. The average queue size is compared to two thresholds, a minimum threshold and a maximum threshold. When the average queue size is less than the minimum threshold, no packets are marked. When the average queue size is greater than the maximum threshold, every arriving packet is marked. If marked packets are in fact dropped, this ensures that the average queue size does not significantly exceed the maximum threshold.

When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with probability \( p \) that is a function of the average queue size.

The general goals have been outlined for congestion avoidance schemes. In this section we describe how the AQM scheme goals (congestion avoidance, no global synchronization and network power) have been met by RED algorithm.

If the RED algorithm in fact drops packets arriving at the gateway when the average queue size reaches the maximum threshold, then the RED guarantees that the calculated average queue size does not exceed the maximum threshold. If its parameters have been set appropriately, then the RED gateway in fact controls the actual average queue size. If the RED algorithm sets a bit in packet headers when the average queue size exceeds the maximum threshold, rather than dropping packets, then the RED relies on the cooperation of the sources to control the average queue size.

The rate at which RED gateways mark packets depends on the level of congestion. During low congestion, the algorithm has a low probability of marking each arriving packet, and as congestion increases, the probability of marking each packet increases. RED algorithm avoid global synchronization by marking packets at as slow a rate as possible.

The RED gateway algorithm could be implemented with moderate overhead in current networks.

The RED explicitly controls the average queue size. Global power (the ratio between throughput and delay) is higher with RED gateways than with Drop Tail algorithm.

The randomized mechanism for marking packets is appropriate for networks with connections with a range of roundtrip times and throughput, and for a large range in the number of active connections at one time. Changes in the load are detected through
Using Median Filtering in Active Queue Management for Telecommunication Networks

changes in the average queue size, and the rate at which packets are marked is adjusted correspondingly. Even in a network where RED algorithm signals congestion by dropping marked packets, there are many occasions in a TCP/IP network when a dropped packet does not result in any decrease in load at the gateway. If the gateway drops a data packet for a TCP connection, this packet drop will be detected by the source, possibly after a retransmission timer expires. If the RED algorithm drops an ACK packet for a TCP connection, or a packet from a non-TCP connection, this packet drop could go unnoticed by the source. However, even for a congested network with a traffic mix dominated by short TCP connections or by non-TCP connections, the RED still controls the average queue size by dropping all arriving packets when the average queue size exceeds a maximum threshold.

The congestion avoidance mechanism should have low parameter sensitivity.

The average queue length is computed by filtering the current queue length. Usually the involved filter is a linear filter with a single coefficient.

The average queue size is limited by a maximum threshold, as long as the calculated average queue size is a fairly accurate reflection of the actual average queue size. The filter coefficient should not be set too low, so that the calculated average queue length does not delay too long in reflecting increases in the actual queue length. The upper bound is required to allow the queue to accommodate bursts of packets without marking packets.

The thresholds should be set sufficiently high to maximize network power. 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.

The threshold difference should be larger than the typical increase in the average queue size during a roundtrip time, to avoid the global synchronization that results when the algorithm marks many packets at one time.

If this difference is too small, then the computed average queue size can regularly oscillate. This behavior is similar to the oscillations of the queue up to the maximum queue size with Drop Tail algorithm.

Often the network traffic is bursty and the linear filtering for average queue length is not efficient. Instead of linear filtering we can use a median filter which remove more efficiently the glitches in the network traffic.

That is, we assume that the traffic is like a noisy signal. If the noise is impulsive then a non-linear filter (like median filter) will have better results than a linear one.

The median filters have been shown to provide improvement in the performance of AQM.

The modified RED algorithm is defined by the following equations:

\[ q(n+1) = (1 - w)q(n) + wq(n+1) \]  
(1)

\[ p(n) = 0, \quad \text{if } q(n) \leq \text{min} \]  
(2)

\[ p(n) = \frac{\text{max} - q(n) \leq \text{min} - q(n) \geq \text{max} \}, \]  
\[ \text{if } q(n) \in (\text{min} \leq q(n) \geq \text{max} \} \]

where \( q \) is the length of the queue, \( \bar{q} \) is the filtered length of the queue, \( p \) is the drop probability of the received packets, \( \text{max} \), \( \text{min} \), and \( \text{max} \) represent the RED parameters for drop probability threshold and minimum and maximum allowed queue length, respectively. The relation (1) represents the linear filter of the queue length and the \( w \) is the filter coefficient.

The median filtering RED scheme replaces the equation (1) with relation (3):
where $M$ represents a odd number and $\text{MEDIAN}$ computes the median value of its parameters.

3. The Main Results

The simulations were made with the following assumptions: the maximum queue length was chosen to be $q_{\text{max}}=200$ packets, the linear filter coefficient is $w=0.02$ and the thresholds were chosen as follows $\text{minth}=50$, $\text{maxth}=150$, $\text{maxp}=0.2$; the median filter length was set to $M=5$.

We have considered some situations in terms of received traffic (that is the received packets from several sources at the gateway input) and transmitted traffic (the transmitted packets to a sink destination). The difference between received and transmitted packets represents the packets that will be load in the queue.

The simulations evaluate the network performance for bursty traffic and the uniform traffic. The drop probabilities of the received packets and the packets delay (or equivalent queue occupancy) were computed. Also the ratio of throughput and delay (called the network power) was calculated.

Figure 1 and 2 illustrate the network performance for a combined traffic (that is, the burst in traffic may occur but there are intervals with uniform traffic as it shows in figures 1a and 2a).

The average drop probability is lower in the median filtering RED algorithm (see figures 1b and 2b) and the “glitches” in the instantaneous drop probability are reduced in the median RED scheme as figures 1d and 2d show. The average queue length (illustrated in figure 1c, figure 2c) and instantaneous queue length (see figure 1e and figure 2e) are similar in both AQM schemes for this types of network traffic.

The network power for the combined traffic is greater for linear filtering RED algorithm comparing with the median filtering RED algorithm, but the difference is quite small as we can see in figure 3.

The queue oscillations may occurs in both schemes but there are smaller in the median filtering AQM scheme.

The case 3 corresponds to a bursty traffic (that is the number of the received packets varies more than transmitted packets). The queue length and the drop probability are increased. The median filtering AQM scheme has a slight better performance; the average drop probability is lower (see figure 4b) and network power is greater (see figure 4d).

The case 4 represents the uniform network traffic (no bursts will occur). The network performance will be enhanced as we can see in figure 5.

The following remarks can be made:

- For a traffic in bursts or combined, the glitches in the drop probability are rejected and the average of drop probability is lower, if the median filter is involved. The network power is similar in both linear and median filtering;
- For a bursty traffic the network power is greater if the median filtering is used;
- For a uniform traffic, there is no significant improvement in performance (drop probability and network power).
Fig. 1. Performance evaluation in case 1:
a) received and transmitted traffic; b) average drop probability; c) average queue occupancy;
d) drop probability and e) queue length.
Fig. 2. Performance evaluation in case 2:

a) received and transmitted traffic; b) average drop probability; c) average queue occupancy; d) drop probability and e) queue length.
Using Median Filtering in Active Queue Management for Telecommunication Networks

Fig. 3. Network power for the cases 1 and 2.

Fig. 4. Performance evaluation in case 3:

a) received and transmitted traffic; b) average drop probability; c) average queue occupancy;

d) network power.
4. Conclusions

This paper evaluates the performance of a median filtering active queue management scheme based on RED algorithm. The median filtering leads to a better network performance in terms of lower drop probability and lower delay from received packets for combined network traffic (bursty and uniform traffic). Also the computational effort is quite reasonable and it does not create any inconvenient considering that the time scale in a network is about hundred of milliseconds or more.

Future work will investigate the influence of the median filter length in the network performance and will analyze a mixed AQM scheme that involve both linear and median filtering depending of the nature of the traffic network.

This paper illustrates how the signal processing techniques will have to contribute, not just at the physical and the application layer, but also in other aspects of network design, and to network science.
References


