```
for each packet arrival
   calculate the average queue size avg
   if min_th ≤ avg < max_th
      calculate probability p_a
      with probability p_a:
         mark the arriving packet
   else if max_th ≤ avg
      mark the arriving packet
```

Figure 1: General algorithm for RED gateways.

empty (the idle period) by estimating the number *m* of small packets that *could* have been transmitted by the gateway during the idle period. After the idle period the gateway computes the average queue size as if *m* packets had arrived to an empty queue during that period.

As *avg* varies from min_{th} to max_{th}, the packet-marking probability p_b varies linearly from 0 to max_p:

$$p_b \leftarrow max_p(avg - min_{th})/(max_{th} - min_{th}).$$

The final packet-marking probability p_a increases slowly as the count increases since the last marked packet:

$$p_a \leftarrow p_b/(1 - count \cdot p_b)$$

As discussed in Section 7, this ensures that the gateway does not wait too long before marking a packet.

The gateway marks each packet that arrives at the gateway when the average queue size *avg* exceeds max_{th}.

One option for the RED gateway is to measure the queue in bytes rather than in packets. With this option, the average queue size accurately reflects the average delay at the gateway. When this option is used, the algorithm would be modified to ensure that the probability that a packet is marked is proportional to the packet size in bytes:

$$\begin{aligned} p_b &\leftarrow max_p(avg - min_{th})/(max_{th} - min_{th}) \\ p_b &\leftarrow p_b\,\mathrm{PacketSize}/\mathrm{MaximumPacketSize} \\ p_a &\leftarrow p_b/(1 - count \cdot p_b) \end{aligned}$$

In this case a large FTP packet is more likely to be marked than is a small TELNET packet.

Sections 6 and 7 discuss in detail the setting of the various parameters for RED gateways. Section 6 discusses the calculation of the average queue size. The queue weight w_q is determined by the size and duration of bursts in queue size that are allowed at the gateway. The minimum and maximum thresholds min_{th} and max_{th} are determined by the desired average queue size. The average queue size which makes the desired tradeoffs (such as the tradeoff between maximizing throughput and minimizing delay) depends on network characteristics, and is left as a question for further research. Section 7 discusses the calculation of the packet-marking probability.

In this paper our primary interest is in the functional operation of the RED gateways. Specific questions about the most efficient implementation of the RED algorithm are discussed in Section 11.

5 A simple simulation

This section describes our simulator and presents a simple simulation with RED gateways. Our simulator is a version of the REAL simulator [19] built on Columbia's Nest simulation package [1], with extensive

```
Initialization:
   avg ← 0
   count ← −1
for each packet arrival
   calculate the new average queue size avg:
      if the queue is nonempty
         avg ← (1 − w_q)avg + w_q q
      else
         m ← f(time − q_time)
         avg ← (1 − w_q)^m avg
   if min_th ≤ avg < max_th
      increment count
      calculate probability p_a:
         p_b ← max_p(avg − min_th)/(max_th − min_th)
         p_a ← p_b/(1 − count · p_b)
      with probability p_a:
         mark the arriving packet
         count ← 0
   else if max_th ≤ avg
      mark the arriving packet
      count ← 0
   else count ← −1
when queue becomes empty
   q_time ← time
```

Saved Variables:

- avg: average queue size
- q_time: start of the queue idle time
- $count$: packets since last marked packet

Fixed parameters:

- w_q: queue weight
- min_{th}: minimum threshold for queue
- max_{th}: maximum threshold for queue
- max_p: maximum value for p_b

Other:

- p_a: current packet-marking probability
- q: current queue size
- $time$: current time
- $f(t)$: a linear function of the time t

Figure 2: Detailed algorithm for RED gateways.

modifications and bug fixes made by Steven McCanne at LBL. In the simulator, FTP sources always have a packet to send and always send a maximal-sized (1000-byte) packet as soon as the congestion control window allows them to do so. A sink immediately sends an ACK packet when it receives a data packet. The