Random Early Detection (RED) Queue Discipline

Mohit P. Tahiliani

Assistant Professor

Department of Computer Science and Engineering

National Institute of Technology Karnataka, Surathkal, India

tahiliani@nitk.edu.in

Overview

- RED is among the first AQM algorithms
  - a.k.a Random Early Discard or Random Early Drop (so there are 3 possible full forms of RED)
  - The main goal of RED is to:
    - avoid congestion
    - avoid the problem of global synchronization
    - avoid the problem of lock out
    - maximize the ‘Power’ function, which is the ratio of throughput to delay
  - Note: RED was proposed in 1993 and the term ‘Bufferbloat’ was coined in 2011.
    - Nevertheless, RED is considered suitable for resolving the issue of Bufferbloat because it controls the ‘queue length’ within specified thresholds.
  - Several variants of RED have been proposed in the literature: Gentle RED, Nonlinear RED (NLRED), Adaptive RED (ARED) and many others.

Working of RED algorithm

- RED operates during the ‘enqueue’ time
  - Note: do not confuse this with ‘input port’ in the router architecture
    - RED runs on the ‘output port’, but during the ‘enqueue’ time!
- RED operates ‘on arrival of every packet’
  - Hence, there is no periodic time interval in which RED is invoked.
  - If packets do not arrive, RED algorithm is not invoked
- RED decides whether the incoming packet should be enqueued or dropped
  - RED algorithm contains the following components:
    - Calculation of average queue length
    - Calculation of drop probability
    - Decision making logic (helps to decide whether the incoming packet should be enqueued or dropped)

Working of RED algorithm (contd ...)

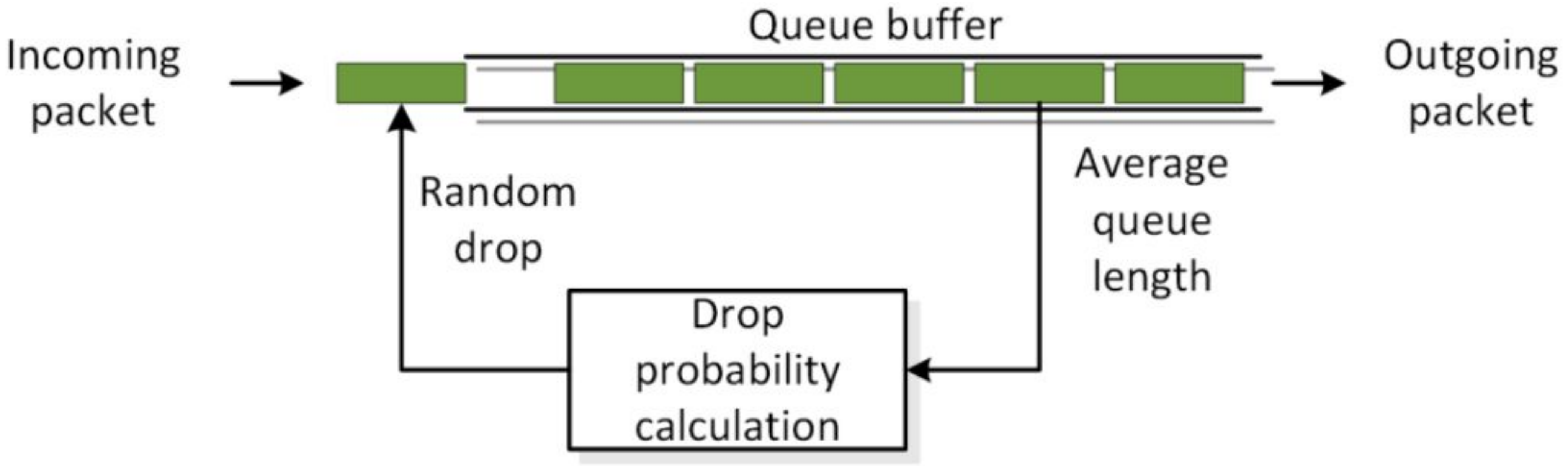

Image Credits: http://ce.sc.edu/cyberinfra/workshops/Material/NTP/Lab%2016.pdf

Working of RED algorithm (contd ...)

1. Calculation of average queue length

   - On arrival of every packet, RED calculates the average queue length using Eq. (1). This mathematical model is known as 'Exponential Weighted Moving Average' or EWMA.

$$newavg = (1 - w_q) \times oldavg + w_q \times current\_queue\_len \qquad \text{Eq. (1)}$$

where,

newavg = new average queue length being calculated in this sample

oldavg = old average queue length obtained during the previous sample

current_queue_len = 'instantaneous' queue length at the router

w_q = weight associated with the 'current_queue_len'. Default value: 0.002

Working of RED algorithm (contd …)

2. Calculation of drop probability

Once the ‘newavg’ is calculated, RED uses the following logic to calculate the drop probability (P_d), where min_{th} represents the minimum threshold for ‘average queue length’ and max_{th} represents the maximum threshold for ‘average queue length’

if (newavg ≤ min_{th})   // default value of min_{th} in the paper is 5 packets

  enqueue the incoming packet // it means $P_d = 0$

else if (newavg > max_{th})   // default value of max_{th} in the paper is 15 packets

  drop the incoming packet // it means $P_d = 1$

else

  Calculate P_d as shown in Eq. (2)

$$P_d = max_p \times [(newavg - min_{th}) \div (max_{th} - min_{th})] \qquad \text{Eq. (2)}$$

where, maxp = maximum drop probability. Default: 0.5 (previously, it was 0.02)

Working of RED algorithm (contd ...)

3. Decision making logic

Once the 'P_d' is calculated, RED uses the following logic to decide whether the incoming packet must be enqueued or dropped:

```
if (P_d ≤ R)
        enqueue the incoming packet
else
        drop the incoming packet
```

where, R = uniformly distributed random number generated between [0, 1]

Note: It is important that a well known random number generator is used to generate R. If the implementation of the random number generator is not correct, RED's performance might get affected.

Working of RED algorithm (contd ...)

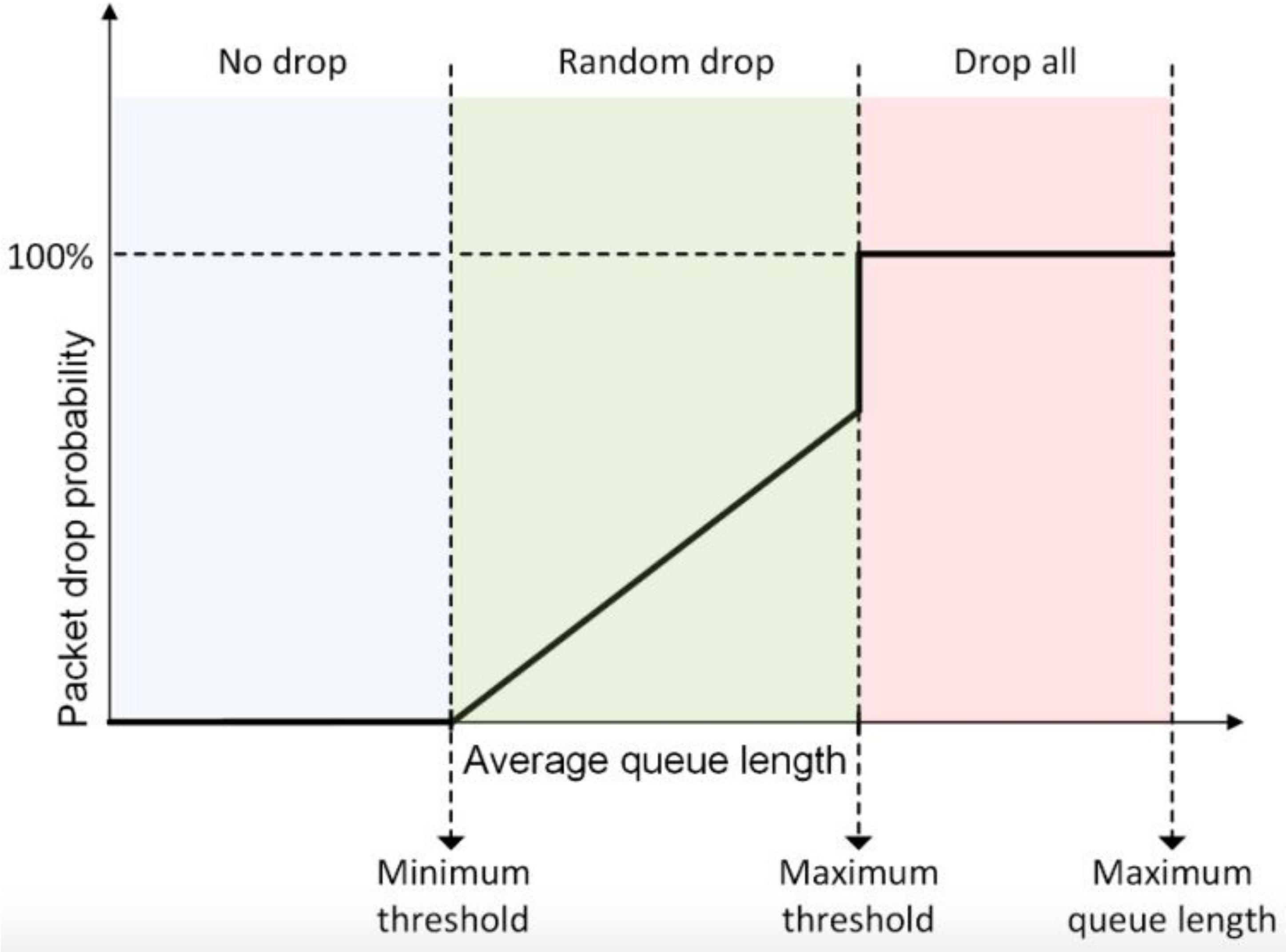

Image Credits: http://ce.sc.edu/cyberinfra/workshops/Material/NTP/Lab%2016.pdf

Working of RED algorithm (contd ...)

Is P_d sufficient to enable an uniform random drop pattern?

- When the average queue size becomes constant, the number of arriving packets 'between' dropped packets becomes a geometric random variable with P_d.
- It implies that packet drops are not uniformly distributed because sometimes more packets arrive and sometimes less packets arrive between two packet drops.
- Hence, a new equation is provided to calculate the drop probability (P_a)

$$P_a = P_d \div (1 - count \times P_d) \qquad \text{Eq. (3)}$$

where, count = number of packets enqueued since last drop

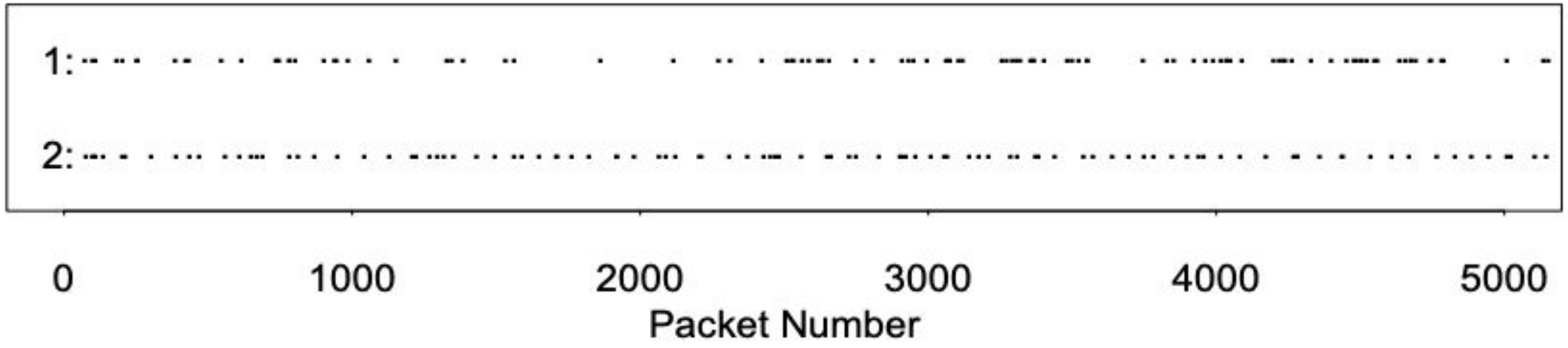

Image Credits: Section 7 in https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/red.pdf

Working of RED algorithm (contd ...)

What happens to 'average queue size' when no packets arrive at the router?

- Since RED is invoked only when packets arrive, there is a possibility that the 'average queue size' value incorrectly indicates heavy congestion even when the queue is actually 'empty'.
    - If this happens, an incoming packet might get dropped even if the queue is totally empty
    - The basic problem is that the value of 'average queue size' does not 'decay' when the queue is idle
- When a first packet arrives at an empty queue, the 'newavg' is calculated as shown in Eq. (4)

$$newavg = (1 - w_q)^m \times oldavg \qquad \text{Eq. (4)}$$

where, m = the number of packets that 'might' have been transmitted by the router if it was not idle.

m is calculated as shown in Eq. (5)

$$m = (idle\_stop\_time - idle\_start\_time) \div (\text{average transmission time of a packet}) \qquad \text{Eq. (5)}$$

where, average transmission time of a packet = mean packet size ÷ bandwidth

- Note: packets can be of different sizes in the network!

Pseudocode for RED algorithm

On arrival of every packet

    if (the queue is empty)

        newavg = $(1 - w_q)^m$ x oldavg

    else

        newavg = $(1 - w_q)$ x oldavg + w_q x current_queue_len

    if (newavg $\leq min_{th}$)

        enqueue the incoming packet

    else if (newavg > max_{th})

        drop the incoming packet

    else

        Calculate P_d and P_a

    if ($P_a \leq R$)

        enqueue the incoming packet

    else

        drop the incoming packet

- Configurable knobs in RED
    a. w_q
    b. min_{th}
    c. max_{th}
    d. max_p
    e. mean packet size

Questions

- How is the upper bound and lower bound calculated for w_q?
    - Hint: See Section 6.1 and 6.2 on: https://www.icir.org/floyd/papers/early.pdf
- What is the rule-of-thumb to configure min_{th} and max_{th}?
    - Hint: See Section 6.3 on: https://www.icir.org/floyd/papers/early.pdf
- What is Weighted RED (WRED)?

Recommended Reading

Random Early Detection Gateways for Congestion Avoidance

Link: https://www.icir.org/floyd/papers/early.pdf

Lab on RED:

Link: http://ce.sc.edu/cyberinfra/workshops/Material/NTP/Lab%2016.pdf