Curvilinear RED

An Improved RED Algorithm

•••

Al Amru Bil Maruf
1805046

Reference

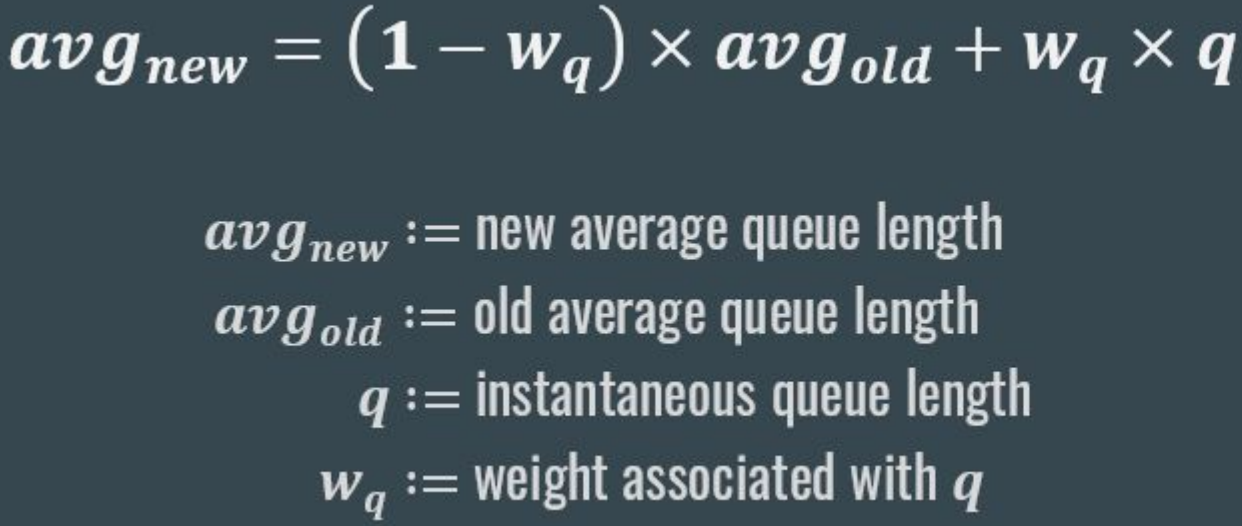
RED: Random Early Detection

An ACM (Active Queue Management) algorithm that implicitly notifies the source of congestion by dropping one or more of its packets.

How it works?

1. Calculate average queue length
2. Calculate drop probability
3. Enqueue or drop a packet

Average Queue Length



$$avg_{new} = (1 - w_q) \times avg_{old} + w_q \times q$$

avg_{new} := new average queue length

avg_{old} := old average queue length

q := instantaneous queue length

w_q := weight associated with q

Drop Probability Function

$$P_d = \begin{cases} 0 & ; avg < min_{th} \\ max_p\left(\dfrac{avg - min_{th}}{max_{th} - min_{th}}\right) & ; min_{th} \leq avg < max_{th} \\ 1 & ; avg > max_{th} \end{cases}$$

max_p := maximum drop probability

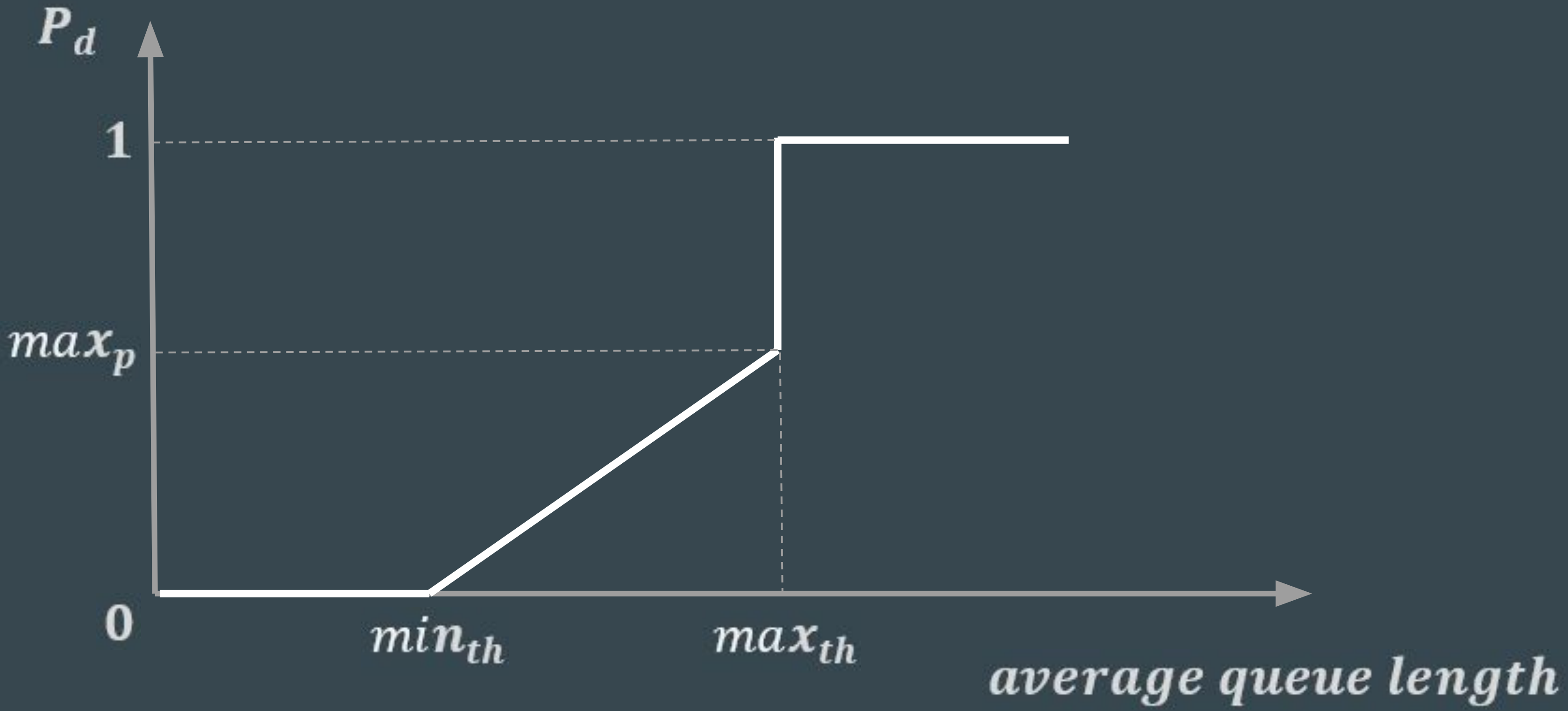
max_{th} := maximum threshold for average queue length

min_{th} := minimum threshold for average queue length

Decision Making

Enqueue or drop packets based on the value of drop probability.

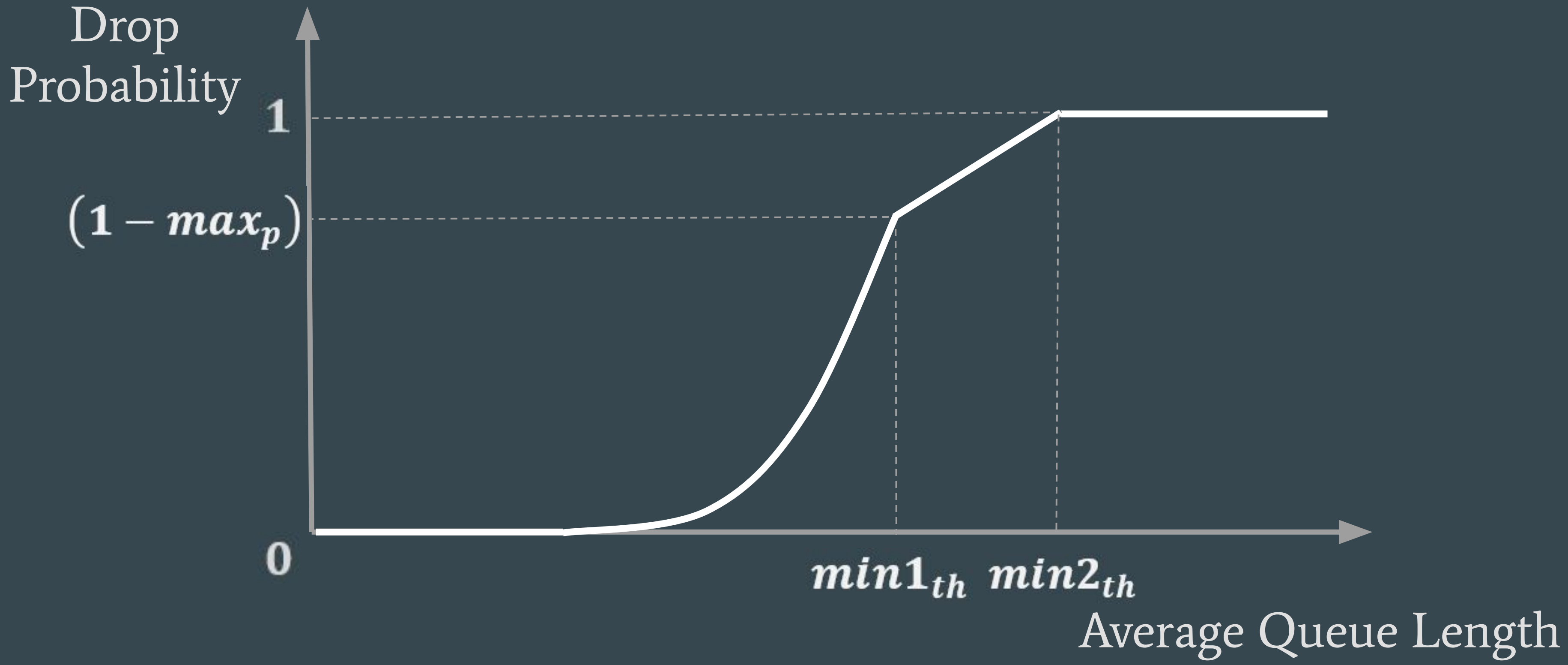
Graph of RED



Curvilinear RED

$$P_d = \begin{cases} 0 & ; avg < min1_{th} \\ 4(1 - max_p)\left(\dfrac{avg - min1_{th}}{max_{th} - min1_{th}}\right)^2 & ; min1_{th} \leq avg < min2_{th} \\ (1 - max_p) + 2max_p\left(\dfrac{avg - min2_{th}}{max_{th} - min1_{th}}\right) & ; min2_{th} \leq avg < max_{th} \\ 1 & ; avg > max_{th} \end{cases}$$

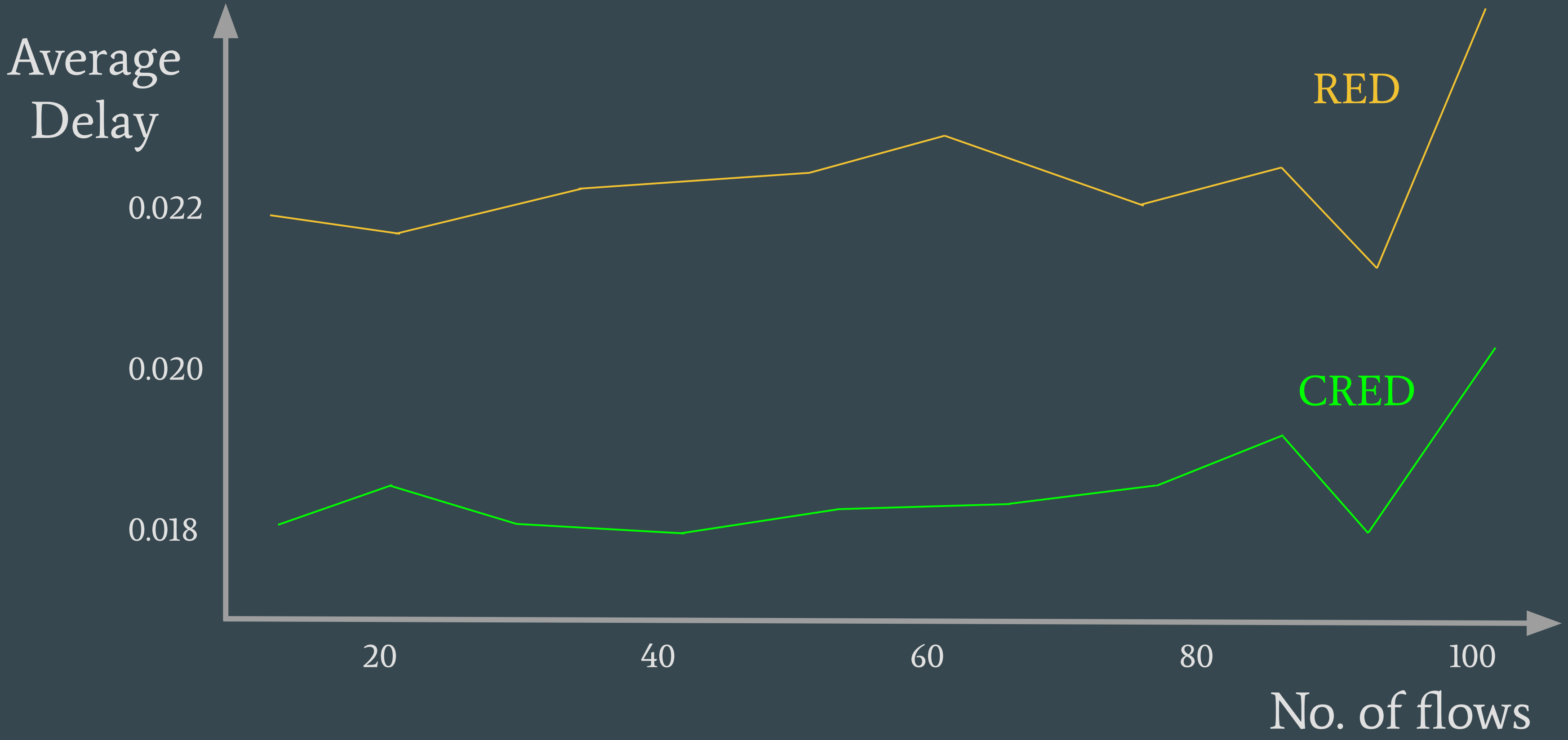
Graph of Curvilinear RED



Curvilinear RED Takeouts

- ➔ Two threshold points instead of one
- ➔ Between the first threshold point and the second, the graph is a quadratic
- ➔ After the second threshold point, the graph is linear

Performance Comparison



Reference

Curvilinear RED: An Improved RED Algorithm for Internet Routers

Authors: Ayodeji Oluwatope, Samuel Hassan

September 2014

DOI: 10.2316/P.2014.813-025

Conference: iasted