Sally Floyd and Van Jacobson

"Random Early Detection Gateway for Congestion Avoidance"

TON, 1993

Router's Queue Management

Manages sharing of

(i) buffer space

(ii) bandwidth

Q1: Which packet to drop (and when to drop) ?

Q2: Which packet to send next?

FIFO + Drop Tail

Keep a single queue

Drop what/when?

Drop arriving packets when queue is full

Send what?

Send the packet at head of queue

Round Robin

One queue per flow

Drop what/when?

Drop arriving packets from flow i when queue i is full

Send what?

Each flow takes turn -- send the packet at the head of the queues in a round robin manner.

Advantages of FIFO and Drop Tail

Simple to implement

Scale well

(no per-connection states)

Reduce delay for a bursty connection
(e.g. VoIP)

Problems with FIFO and Drop Tail

Problem 1
Bias against bursty traffic

burstiness increases chances that the queue will overflow

Problem 2
Global synchronization

connections reduce their windows simultaneously, lowering utilization.

Problem 3
Queue size

higher bandwidth needs longer queue, increasing delay.
TCP tries to keep the queue full.

Problem 4

No isolation against unresponsive flows

Random Drop

Keep a single queue

Drop what/when?

Drop random packet in the queue when queue is full

Send what?

Send the packet at head of queue

No bias against bursty traffic -- bursty arrival causes random packets to be dropped.

Flows with higher rate occupy more buffer spaces, have higher chance to be dropped.

Signal flows that is congesting the network to slow down.

Random drop recovers from congestion (full queue) by dropping packets.

Early Random Drop

Drop what/when?

Drop arriving packet randomly when queue is longer than a threshold

Early random drop avoids congestion (full queue) by dropping packets before queue is full.

RED

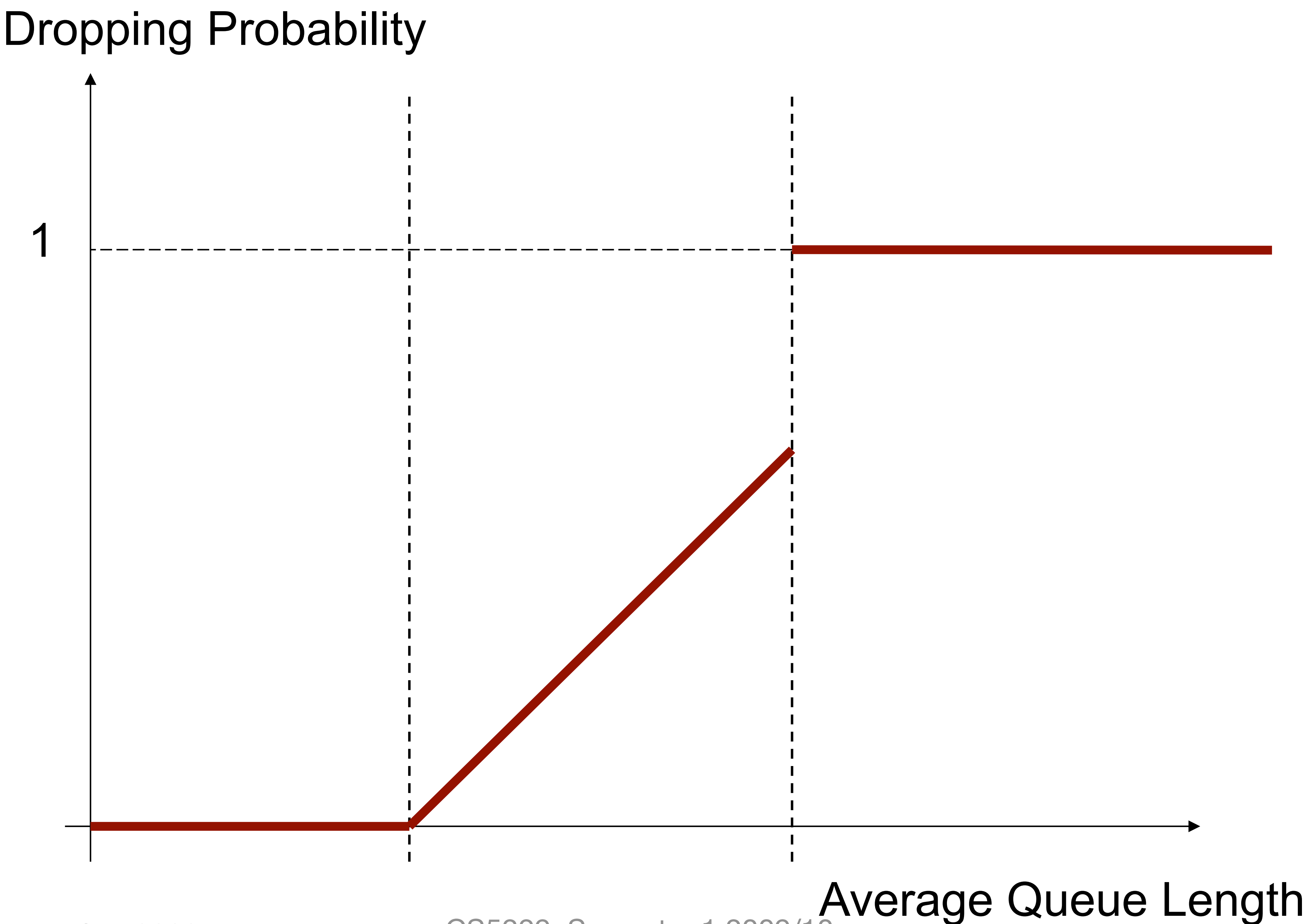
Random Early Detection

Drop what/when?

Drop arriving packet randomly when average queue length is above a threshold

Differences: Use average queue length instead of instantaneous length to absorb transient congestion.

Differences: Dropping probability changes dynamically depending on queue length.


Dropping Probability
1
Average Queue Length

foreach incoming packet X

calc average queue length

if min_{th} < average < max_{th}

calc p

drop X with probability p

else if average > max_{th}

drop X

(Instead of dropping packets, we can also mark a packet to indicate congestion)

How to calculate average queue length?

How to calculate drop probability?

How to set thresholds?

We can use exponentially weighted average. On every packet arrival:

$$avg \leftarrow (1 - w_q)avg + w_q q$$

Large w_q : A burst of packets will cause avg to increase too fast, hit the max threshold

Small w_q : avg increases too slowly and we are unable to detect initial stage of congestions.

We can use exponentially weighted average. On every packet arrival:

$$avg \leftarrow (1 - w_q)avg + w_q q$$

What if q drops to zero and no packet arrives?

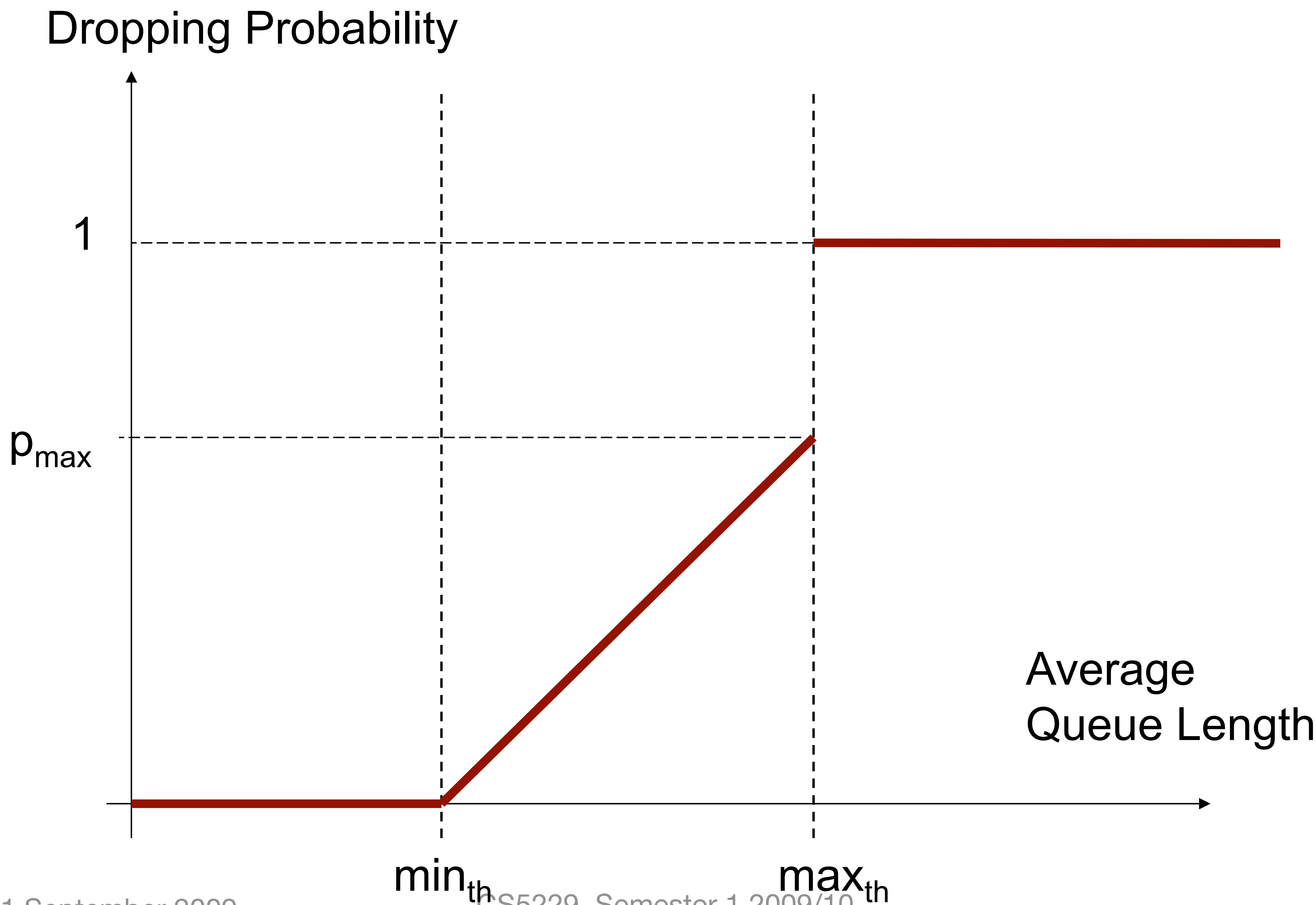
$$avg \leftarrow (1 - w_q)^m avg$$

m is a function of period when queue is empty

How to calculate average queue length?

How to calculate drop probability

How to set thresholds?


Dropping Probability
1
p_{max}
Average
Queue Length
min_{th}
max_{th}

How to calculate average queue length?

How to calculate drop probability?

How to set thresholds?

max_{th} - min_{th} should be sufficiently large otherwise average queue size can oscillate beyond max_{th}

“need more research” for optimal value.

Advantages of RED

No bias against bursty flows

Less global synchronization

Control average queue length

RED does not deal with unresponsive flows

ns-2 demo

Variations of RED

RED biases against flow with **small** packet size

We can fix this by weighting drop probability to packet size

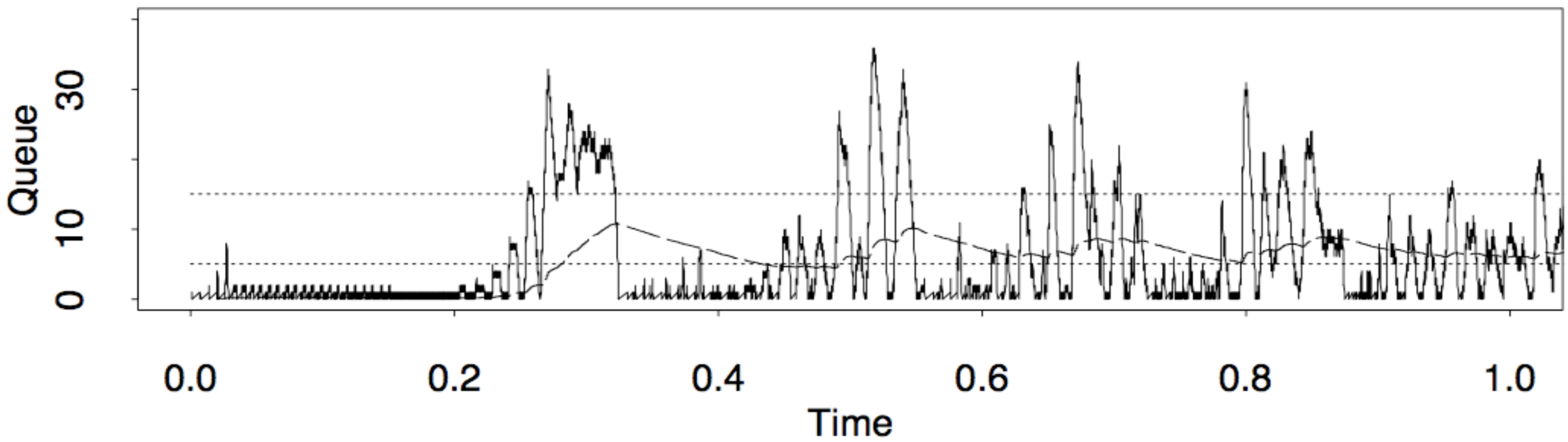
A router can keep one queue per flow and apply RED to each one.

Drop probability can be weighted with the priority of the flow.

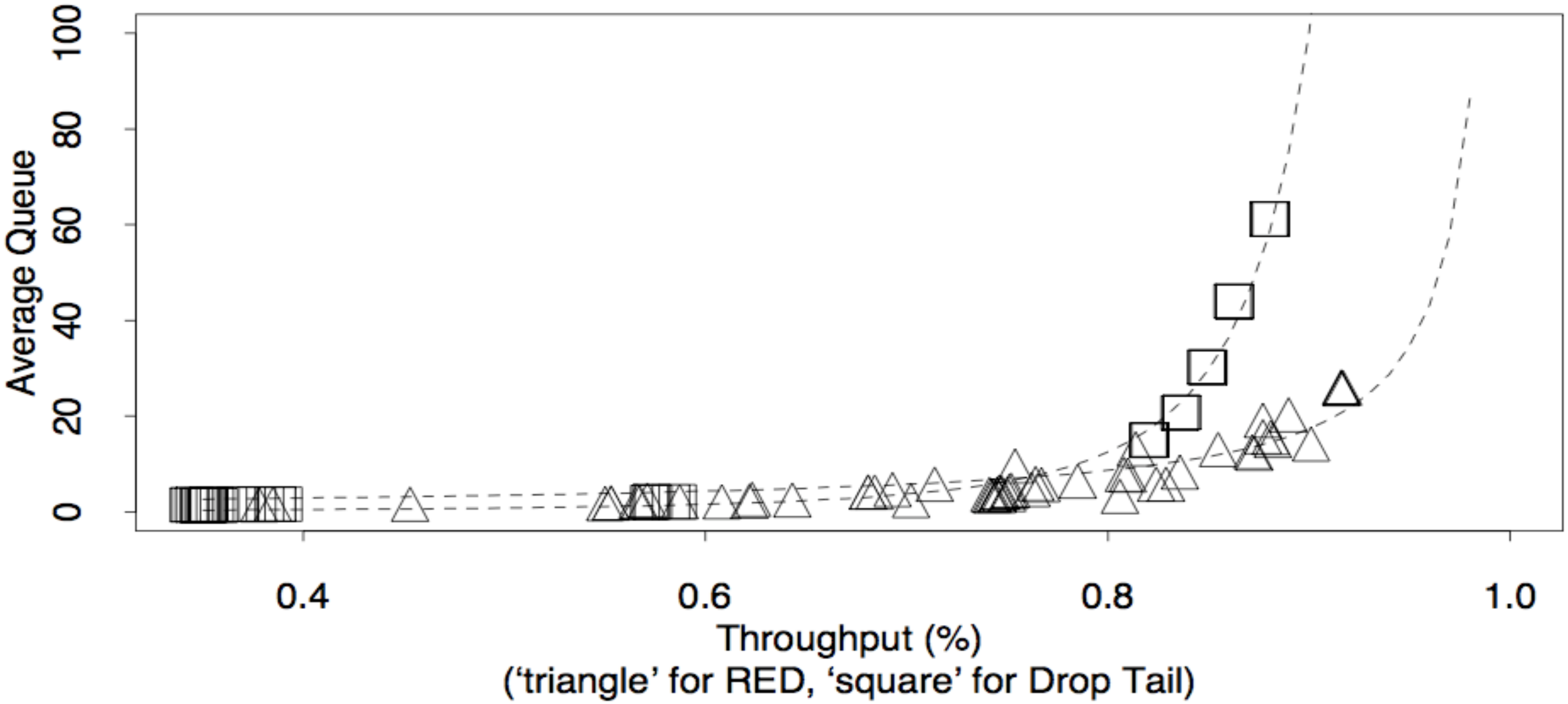
This is known as WRED and is implemented in some Cisco routers.

Simulation Results

Four TCP flows starting at time 0.2, 0.4, 0.6 and 0.8



Queue size (solid line) and average queue size (dashed line).



Average Queue
0
20
40
60
80
100
0.4
0.6
0.8
1.0
Throughput (%)
('triangle' for RED, 'square' for Drop Tail)

Conclusion:

RED increases throughput, reduces delay, controls average queue sizes, reduces global sync, and is fairer to bursty traffic. It is deployed in routers today.

But careful tuning of parameters is needed.