Router's
Queue Management

Manages sharing of
(i) buffer space
(ii) bandwidth

Q1: Which packet to drop when queue is full?

Q2: Which packet to send next?

FIFO + Drop Tail

Keep a single queue

Answer to Q1:
Drop arriving packets
when queue is full

Answer to Q2:
Send the packet at
head of queue

Round Robin

One queue per flow

Answer to Q1:
Drop arriving packets from
flow i when queue i is full

Answer to Q2:
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
Baised againts bursty traffic

burstiness increases changes that the queue will overflow

Problem 2
Global synchronization

connection reduces 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

Answer to Q1:
Drop random packet in the queue when queue is full

Answer to Q2:
Send the packet at head of queue

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

Flows with higher rate occupies more buffer spaces, have more 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

Answer to Q1:
Drop arriving packet randomly if queue is longer than a threshold

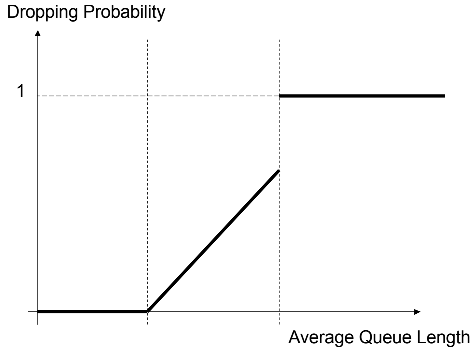
Random drop avoid congestion (full queue) by dropping packets before queue is full.

RED
Random Early Detection

Answer to Q1:
Drop arriving packet randomly if average queue length is above than a threshold

Differences 1: Use average queue length instead of instantaneous length to absorb transient congestions.

Differences 2: Dropping probability should change dynamically depending on 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 set the ECN bit 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:

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

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?

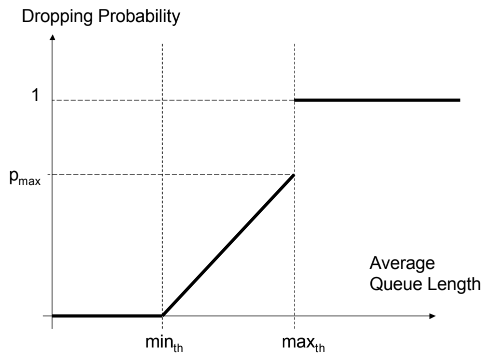
$$avg \leftarrow (1 - w_q)^m avg$$

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?



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 average queue size.

Advantages of RED

No bias against bursty flows
Less global synchronization
Control average queue length

Variations of RED

RED does not deal with unresponsive flows

RED biases against flow with large packet size

We can fix this by weighting drop probability to packet size

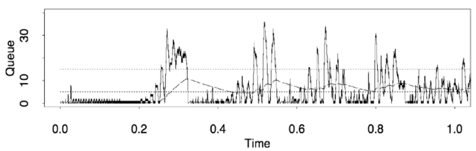
A router can keep one queue per flow and apply RED to each one.

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

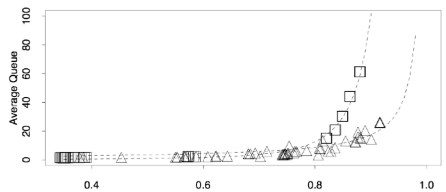
This is known as WRED and is implemented in some Cisco routers.

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).



('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.