Random Early Detection (RED) gateways

Sally Floyd
CS 268: Computer Networks
floyd@ee.lbl.gov

March 20, 1995

The Environment

- Feedback-based transport protocols (e.g., TCP)

Problems with current Drop-Tail gateways

- If queue size (buffer allocated for queue) is small, difficulties for large-window TCP connections during slow-start and little ability to accomodate transient congestion. If queue is large, unnecessary delay for delay-sensitive traffic (e.g., telnet TCP).

- Global synchronization, resulting in reduced aggregate throughput [Zhang and Clark, 1990].

- Biases against bursty traffic [Floyd and Jacobson, 1992].

- Fairness concerns, phase effects, etc.

Underlying problems of Drop-Tail gateways

- Transient and persistent congestion are not distinguished.

- Gateway has no control over packet drops (congestion feedback).

Random-Drop gateways give the gateway some control of *which* packet to drop, but do not take care of all of the problems of Drop-Tail gateways.

Approach of RED gateways

- Distinguish between transient and persistent congestion. Design the network to accomodate bursty traffic, rather than shaping bursty traffic to accomodate the needs of the network.

- The most effective detection of persistent congestion can occur at the gateway.

- Let the gateway control packet drops.

- When possible, use FIFO scheduling within a traffic class. FIFO has low overhead, no scaling problems, and reduces the tail of the delay distribution.

- Allow for gradual deployment, flexibility, and evolution.

RED gateway algorithm:

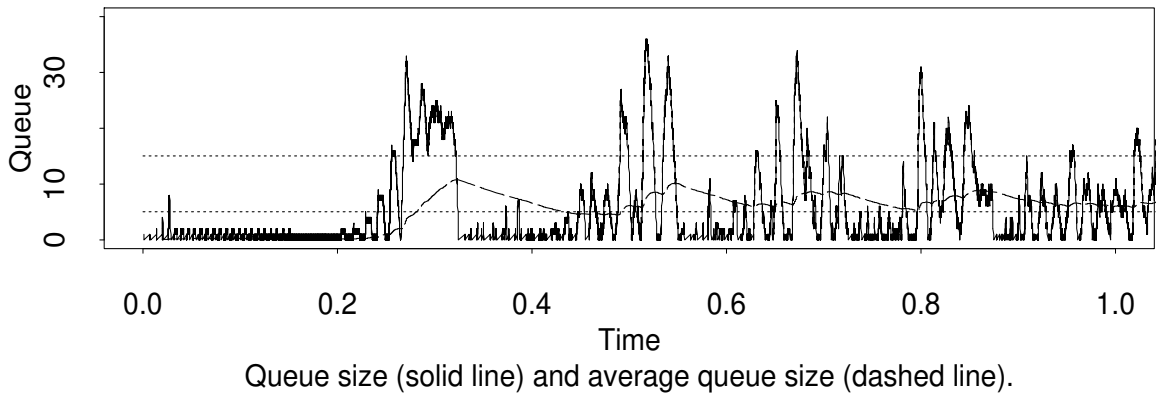
for each packet arrival
    calculate the new average queue size avg
    if $min_{th} \leq avg < max_{th}$
        calculate probability p_a
        with probability p_a:
            mark/drop the arriving packet
    else if $max_{th} < avg$
        drop the arriving packet

Variables:

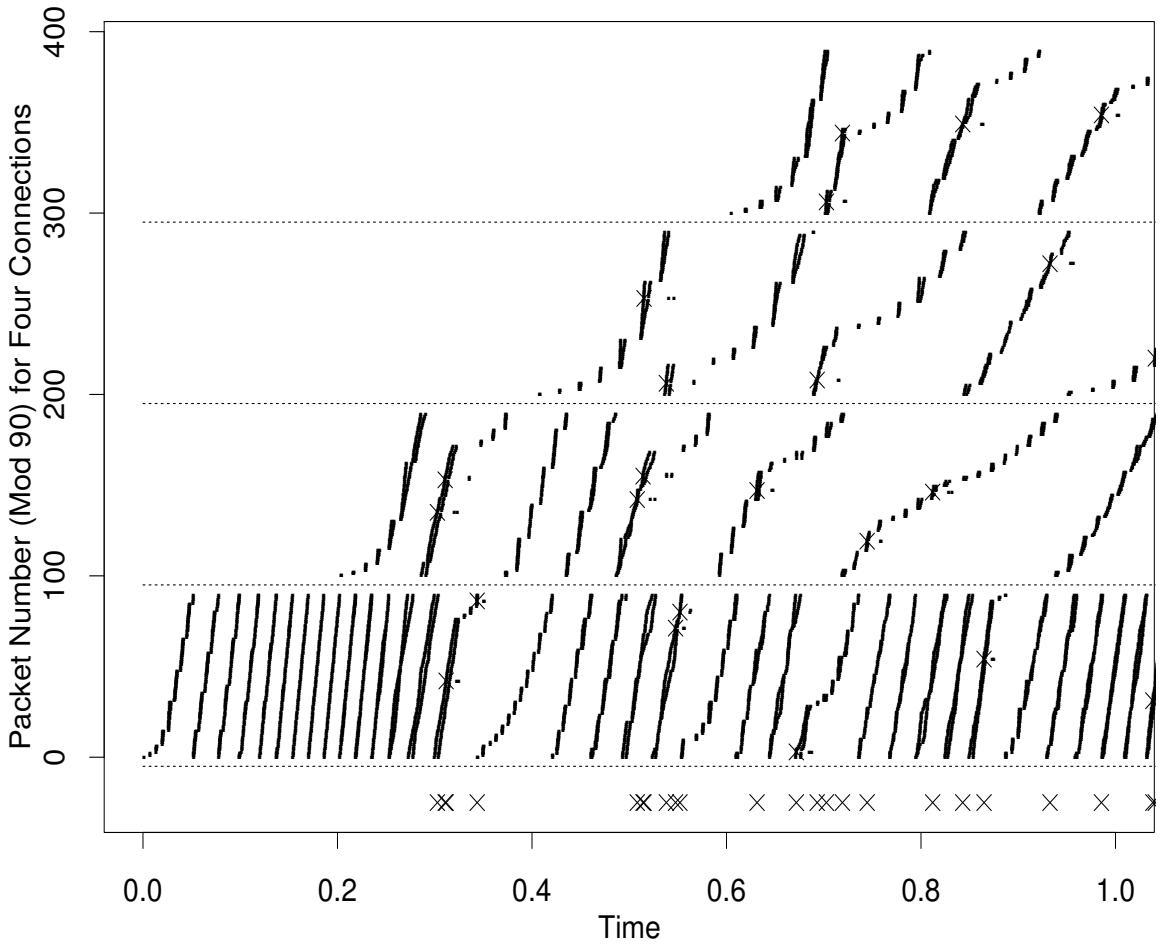
avg: average queue size
p_a: packet marking/dropping probability

Fixed parameters:

min_{th}: minimum threshold for queue
max_{th}: maximum threshold for queue



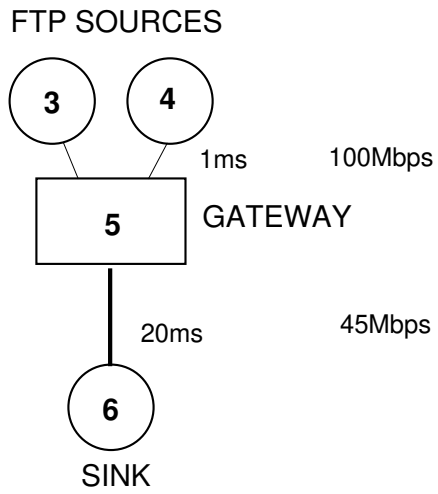
Queue size (solid line) and average queue size (dashed line).

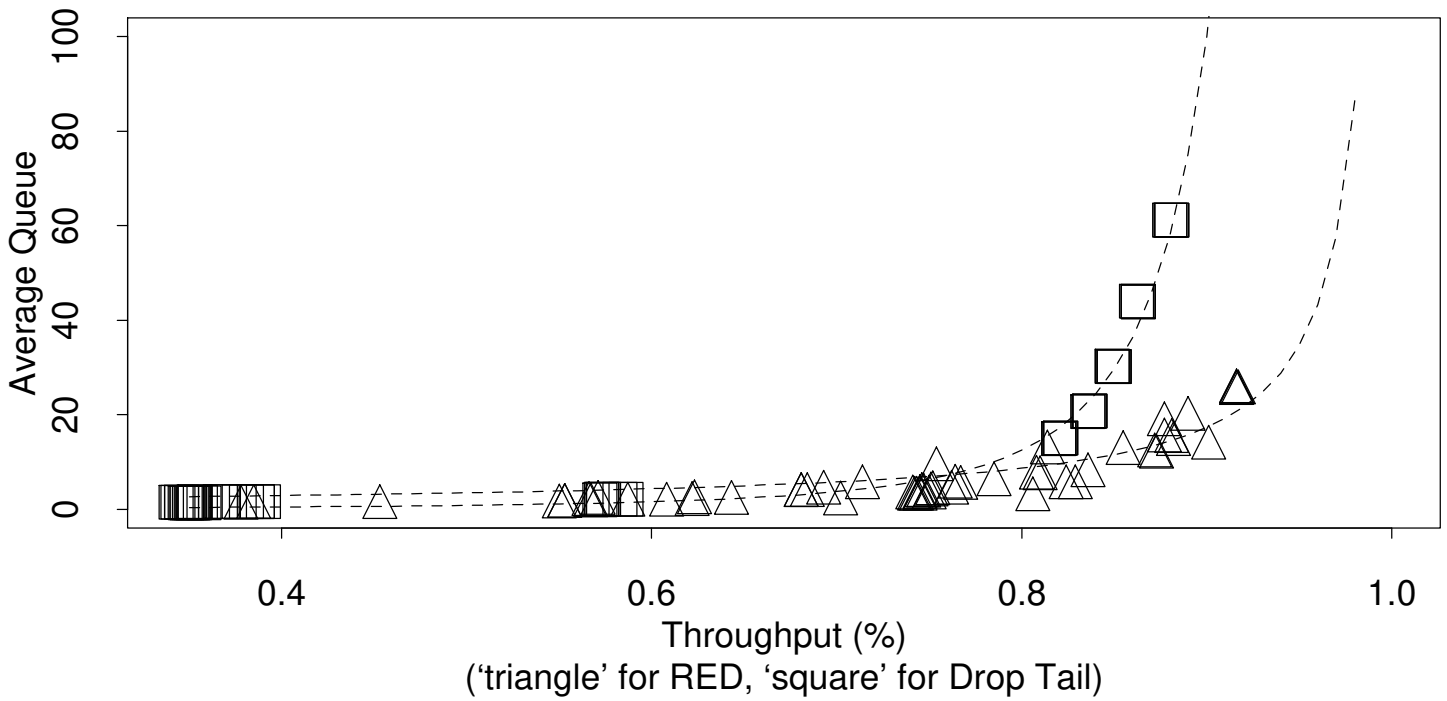


A simulation with four FTP connections with staggered start times.

Comparing Drop Tail and RED gateways.

Goals: high throughput, low average delay.





Calculating the average queue size

for each packet arrival
  if the queue is nonempty
    $avg \leftarrow (1 - w_q)avg + w_q\, q$
  else using a table lookup:
    $avg \leftarrow (1 - w_q)^{(time - q_time)/s} avg$

Parameters:

w_q: queue weight
s: typical transmission time
q: current queue size
$time$: current time
q_time: start of the queue idle time

For w_q a power of two (e.g., $2^{-9} \approx 0.002$), this is efficient to implement.