Lecturer: Dr. Michael J. May
Kinneret College

Fair Queuing, RED

3 February 2025
Lecture 13

Topics for Today

- Fair Queuing
- Congestion Avoidance
  - RED

- Sources: PD 4.1.2, KR 4.6.3

Queuing Techniques

First In First Out (FIFO)

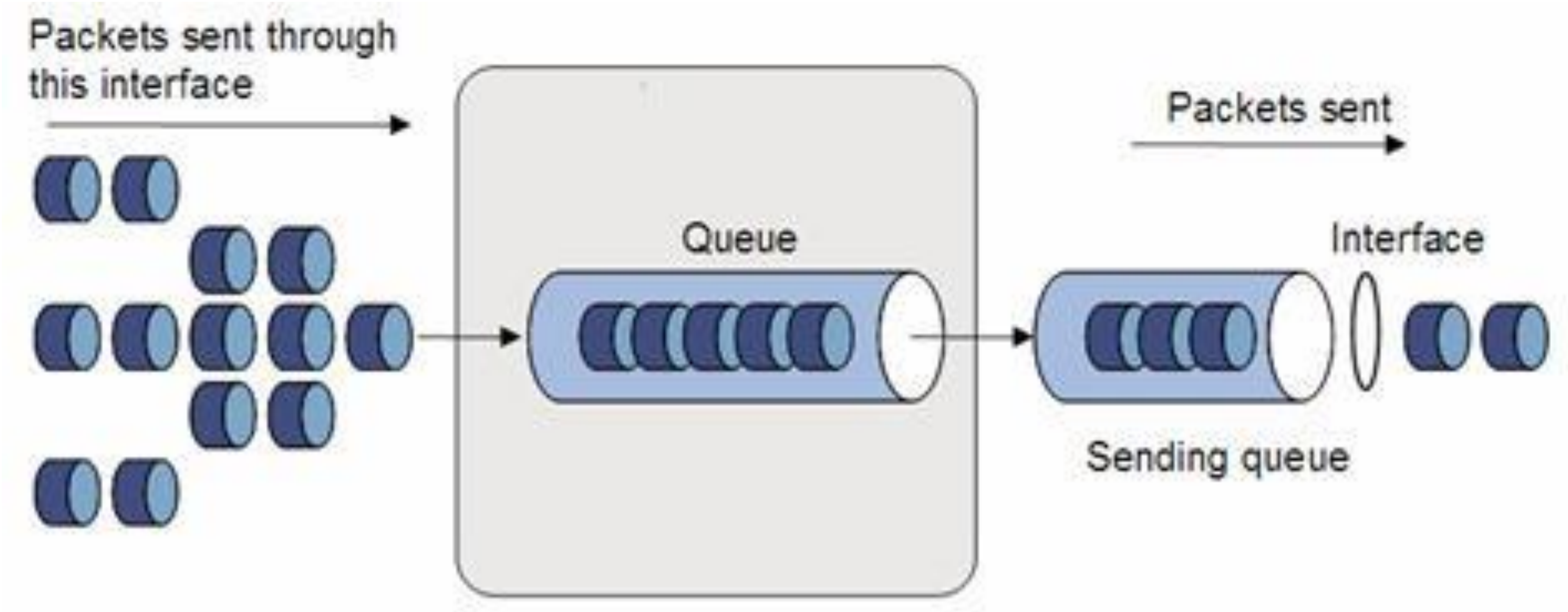
Priority Queuing (PQ)

Fair Queuing (FQ)

Weighted Fair Queuing (WFQ)

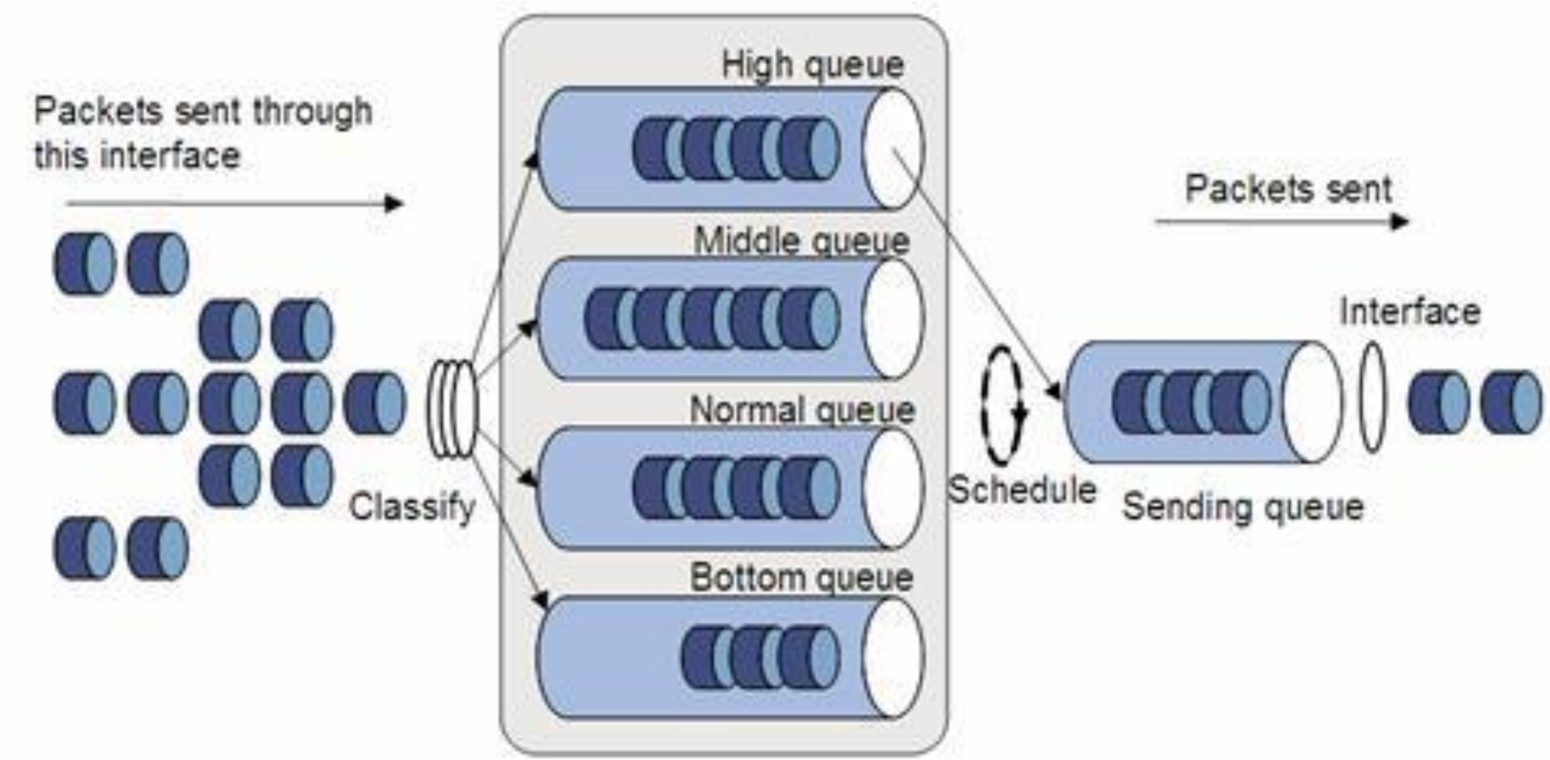
First In First Out

- Rule: Packets are sent out of the router as they arrive



Priority Queuing

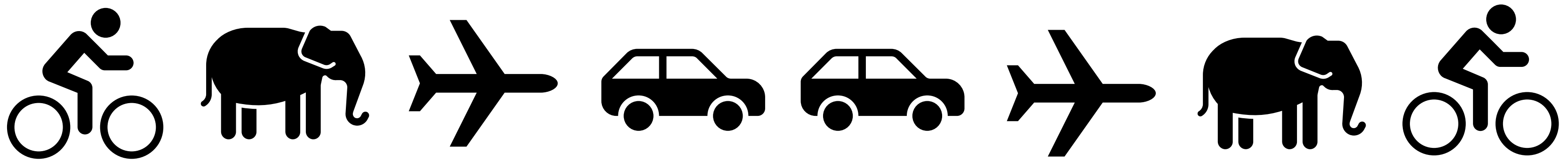
- Put a strict order on the queues
  - Highest priority first, then secondary ones
  - Advantages? Disadvantages?



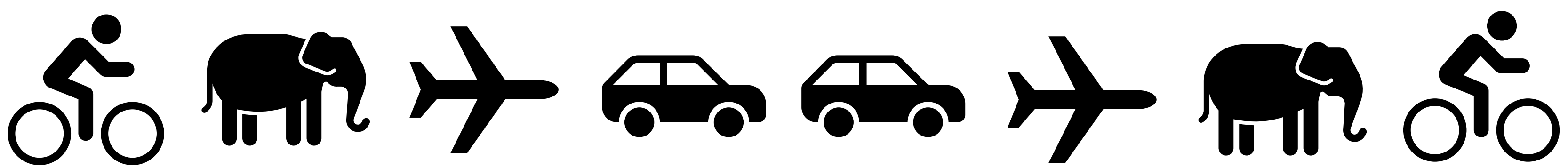
https://money.cnn.com/2016/05/18/pf/blame-for-airport-security-lines/index.html

Queuing Options: FIFO

In

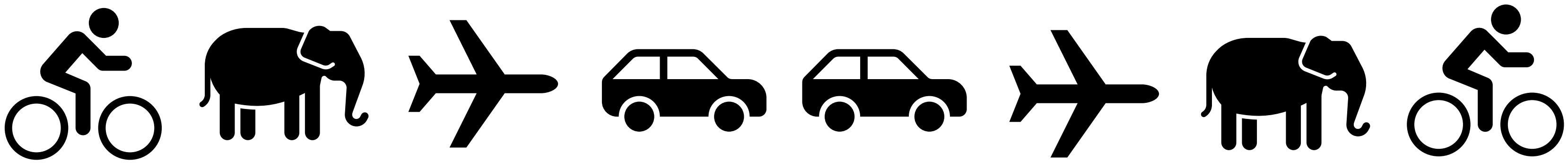


Out

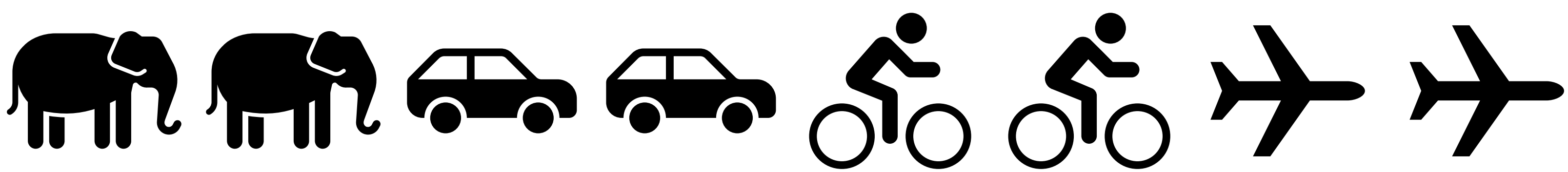


Queuing Options: Priority

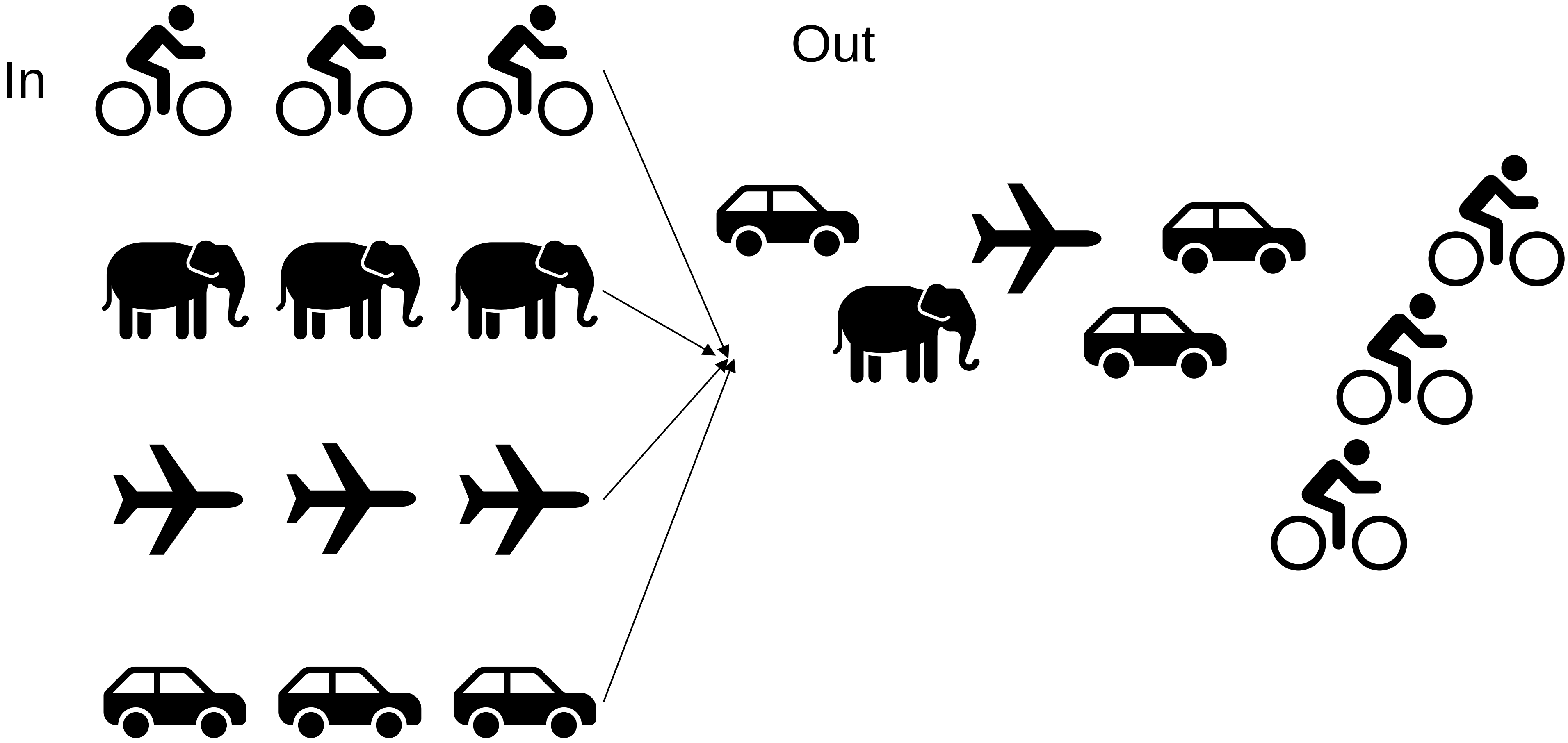
In



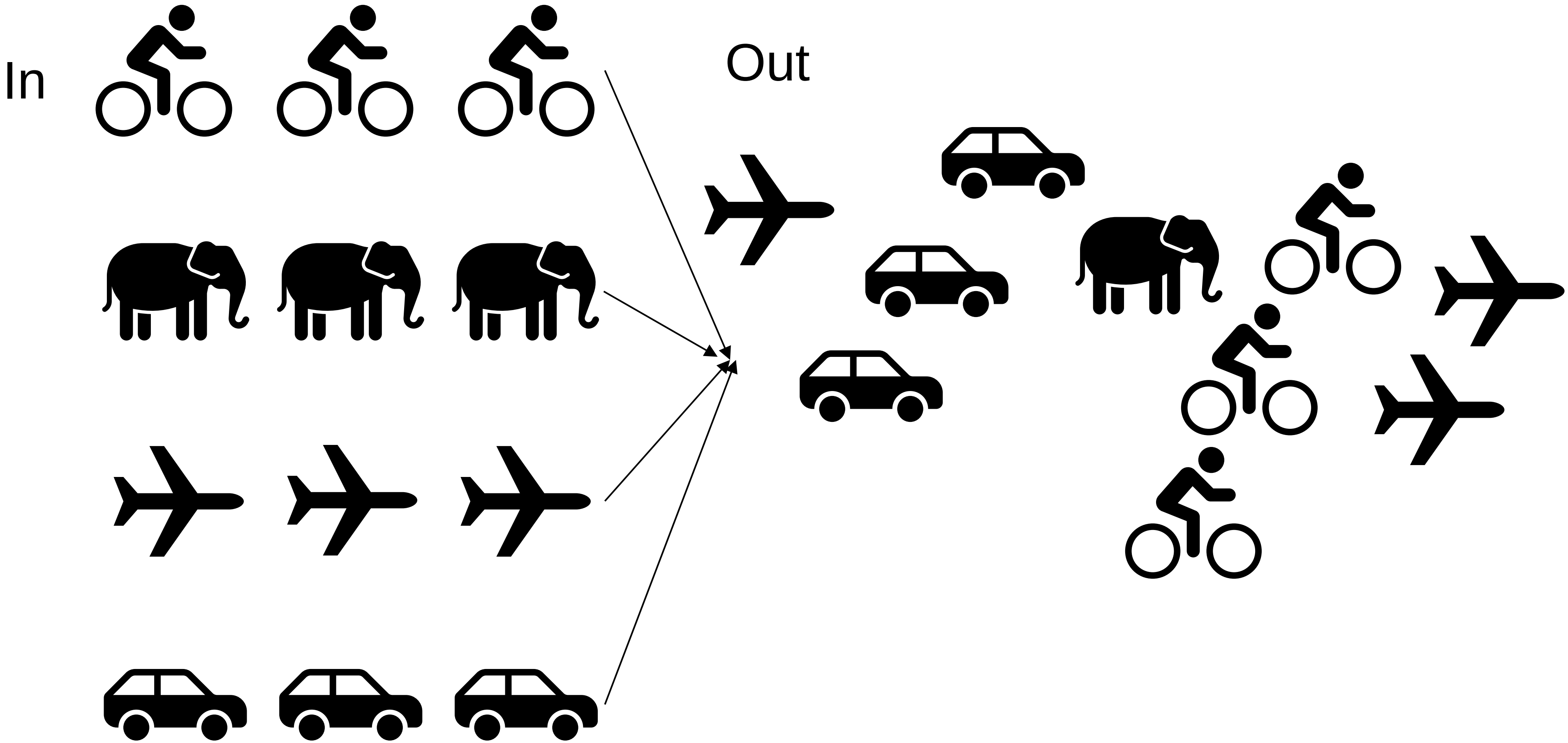
Out



Queuing Options: Fair



Queuing Options: Weighted Fair



Fair Queuing

FIFO can be overrun by an out-of-control sender

- Router can intervene to make things fairer

Fair Queuing:

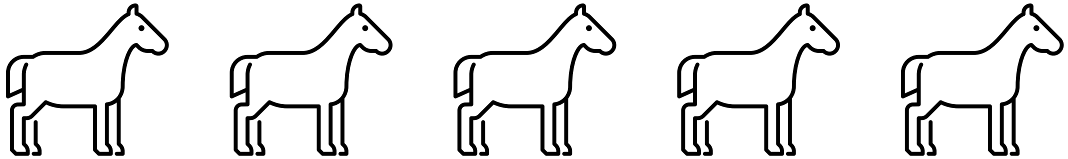
- Give each flow a queue
- Manage each flow separately and service them Round Robin
- If one flow's queue is full, we need to drop (somehow)
- Each queue gets to send one packet at a time, but we don't interrupt

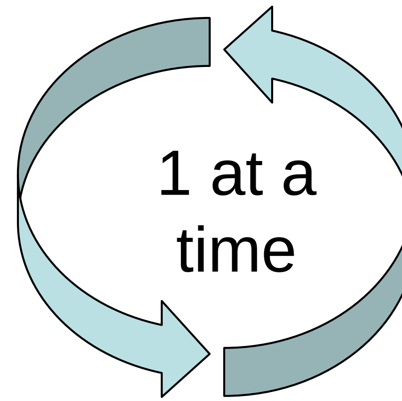
What if one sender sends $1000B$ packets and another sends $500B$ packets?

- To be fair: We want a $1000B$ packet to "cost" as much as two $500B$ packets

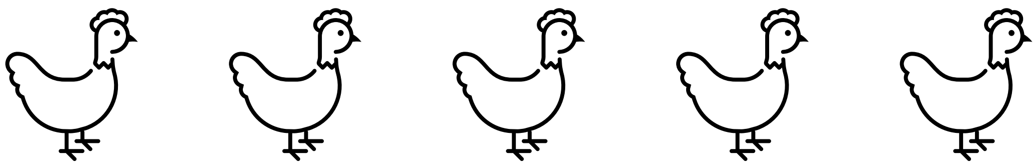
Fairness?

Flow 1

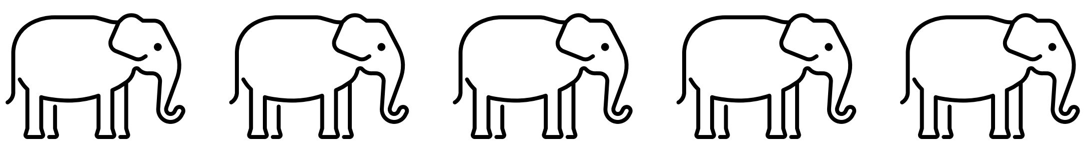




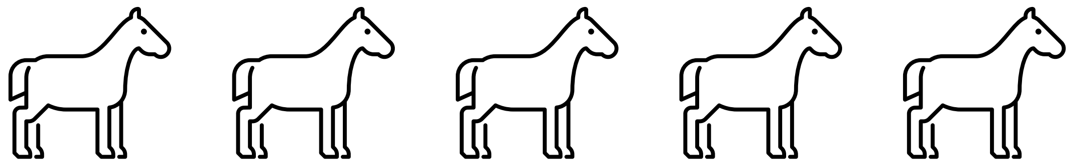
Flow 2

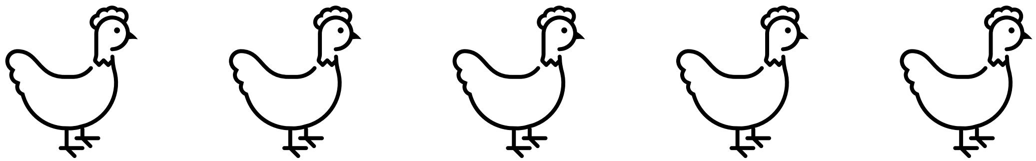


Flow 3



Fairness? Divide it in 3!

Flow 1  

Flow 2  

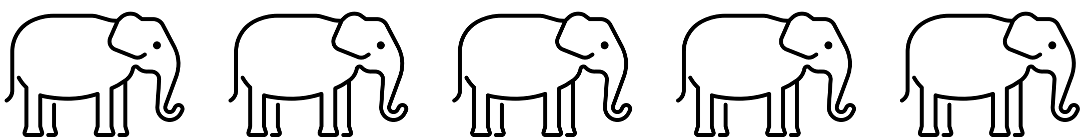
Flow 3  

Image © Marie-Lan Nguyen / Wikimedia Commons

No Interruptions

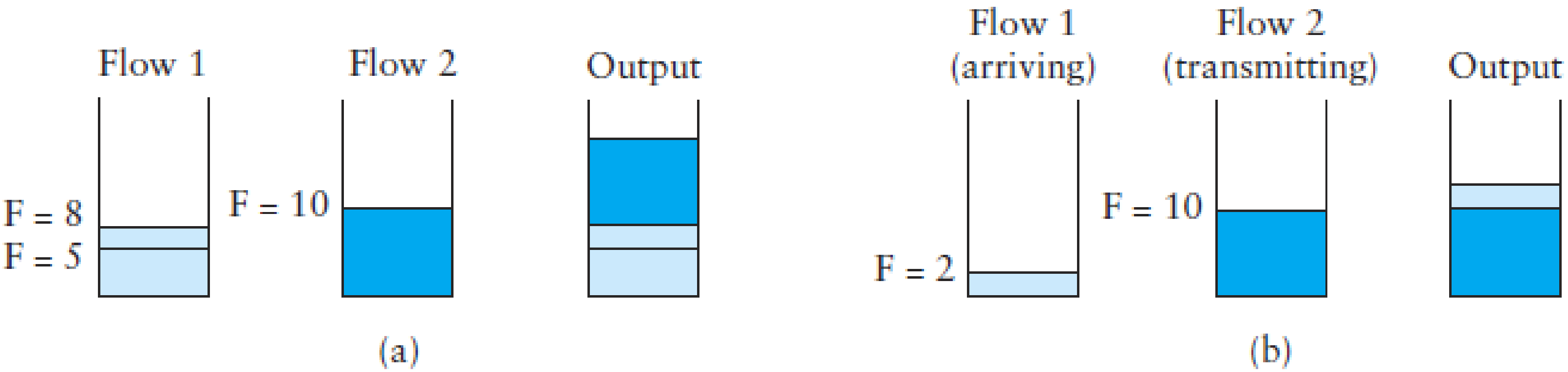


Figure 6.7 Example of fair queuing in action: (a) shorter packets are sent first; (b) sending of longer packet, already in progress, is completed first.

Fair Queuing Theory

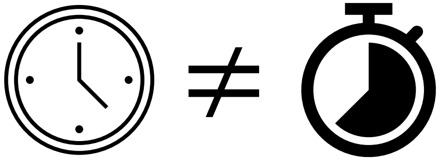
- For each packet:
  - Imagine there are no other flows on the router
  - Determine when it *would have finished being sent* based on when it arrived – save it with the packet in the queue
  - Think that a $500B$ packet takes 500 "ticks" to send

- Packet i arrives at time A_i, it has size P_i, and begins being sent at S_i
  - Then it finishes being sent at $F_i = S_i + P_i$
- What is S_i?
  - Case 1: There is another packet F_{i-1} from the flow being sent on the line – then $S_i = F_{i-1}$
  - Case 2: The line is free – then $S_i = A_i$

- Result: $F_i = \max(F_{i-1}, A_i) + P_i$

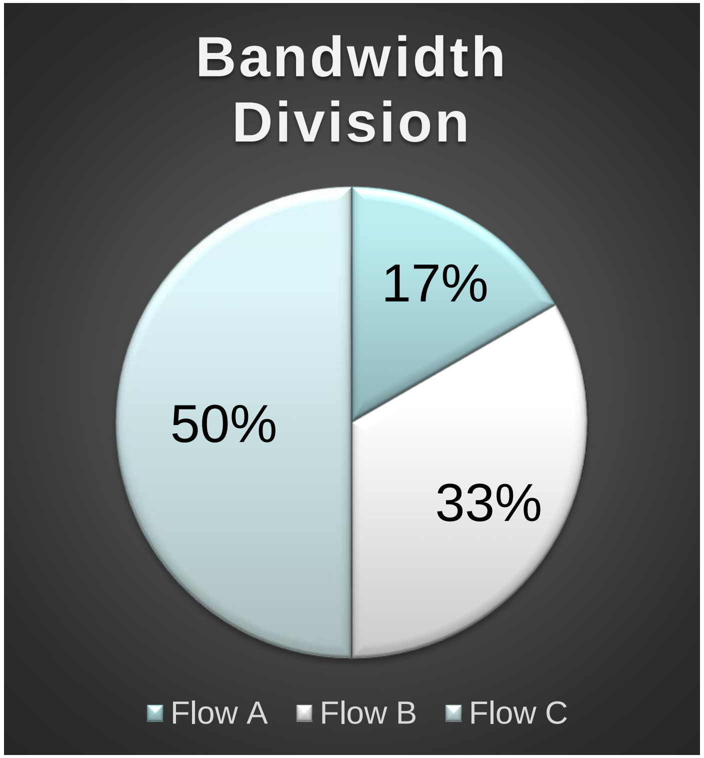
Fair Queuing

- But there are other flows on the router too
  - The “tick” should be when *each flow had a chance to send 1 byte*
  - So slow down the clock calculations for A_i based on the number of active queues
  - If there are 3 active queues, A increments in 0.333 instead of 1
  - This means that the A doesn’t match the real “wall clock” time



Weighted Fair Queuing

- Give a *weight* to each flow to give it more of the bandwidth
- Flow A gets 1, Flow B gets 2, Flow C gets 3
  - Flow A has $\frac{1}{6}$ of the flow
  - Flow B has $\frac{2}{6}$ of the flow
  - Flow C gets $\frac{3}{6}$ of the flow



- Calculate by dividing the size P_i by the weight of the flow
  - Packet $A1 = 300B = \ 300$ ticks
  - Packet $B1 = 300B = \frac{300}{2} = 150$ ticks
  - Packet $C1 = 300B = \frac{300}{3} = 100$ ticks

Queuing Examples

So Far

- Fair Queuing
- Congestion Avoidance
  - RED

Congestion Avoidance

- Congestion occurs when there are too many packets and not enough bandwidth/space in the pipe to fit them

- What if we could prevent congestion in the first place?
  - Tell the senders to slow down so we never need to drop packets at all?

- How could we do this?

Explicit notification	Implicit notification

Explicit Notification

Implicit Notification

https://www.someecards.com/usercards/viewcard/bb698ec1f18238c01fcbd9f86b272f06af/

Explicit Notification: DECbit

- Designed for Digital Network Architecture
  - Connectionless with a connection-oriented transport protocol
    - Just like TCP/IP

- Add a bit to the header – the congestion bit
  - Router sets it when it sees its queue lengths are too long
    - If the average queue length is greater than 1 over the last busy/idle interval
  - The receiver copies the bit to the acknowledge field

The sender measures the percent of arriving packets with the congestion bit set over the last send window

- If $> 50\%$ have it set, reduce congestion window to 87.5%
- If $< 50\%$ have it set, add 1 packet to the congestion window
  - AIMD!

Implicit Notification: RED

Observation: When TCP sees that a packet has dropped, it lowers the congestion window to half

- We can use that to “advise” TCP senders to reduce their sending speed before things get too bad

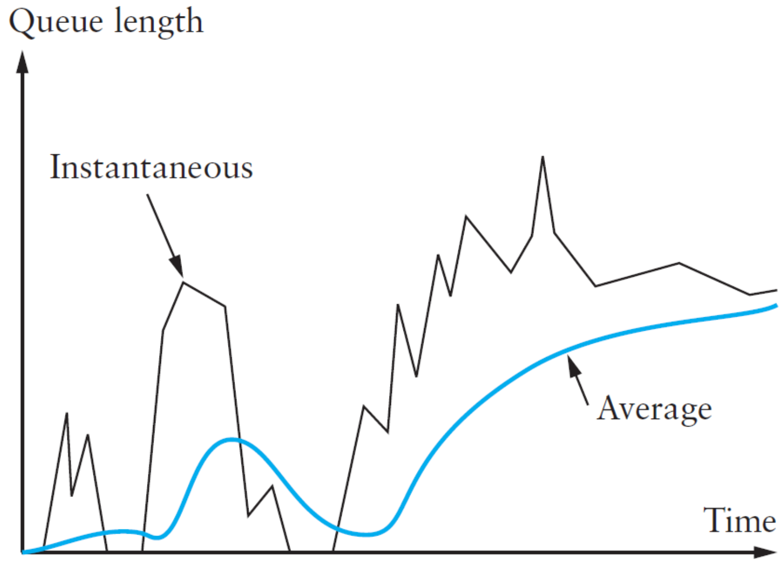
The details:

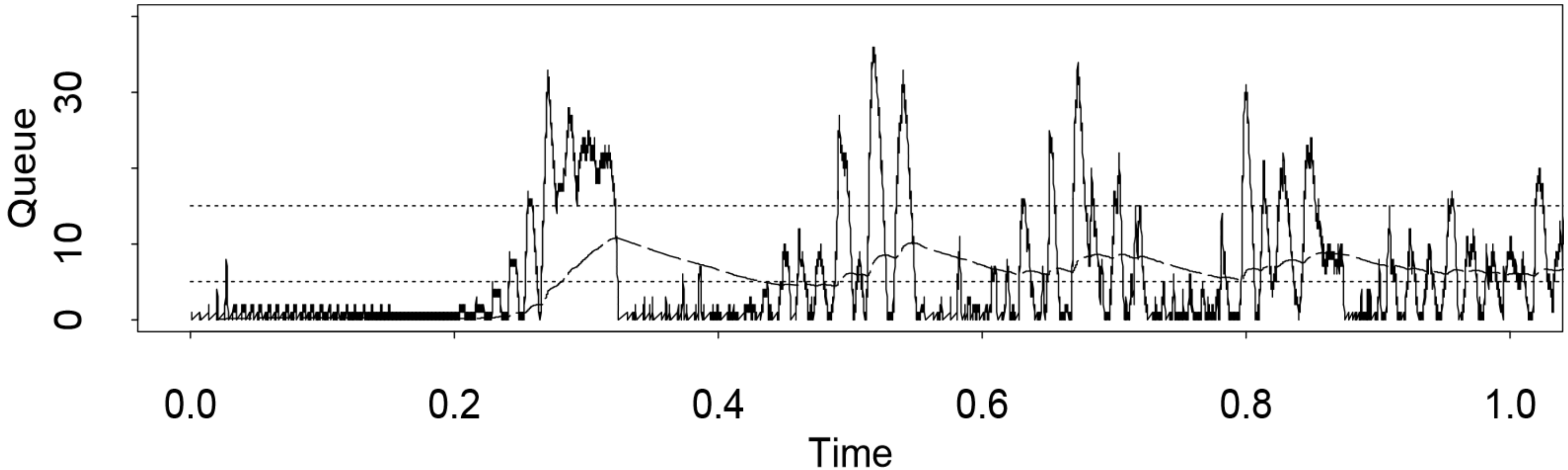
- Router tracks the average queue length to decide
  - $AvgLen = (1 - Weight) \times AvgLen + Weight \times SampleLen$
  - Choose $Weight$ to balance new vs. old state
    - Takes at least $1\ RTT$ for drop to have effect
  - Recommendation [Floyd and Jacobson, 93]:

$$0.001 \le Weight \le 0.0042$$

for $MinThreshold = 5$ and bursts of up to 50

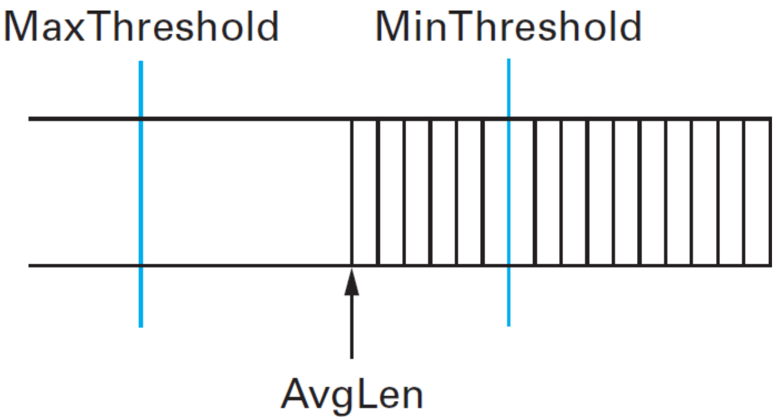
Smoothing Queue Length





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

RED Algorithm



Choose a $MinThreshold, MaxThreshold$

When a packet arrives:

1. Recalculate $AvgLen$
2. If $AvgLen \leq MinThreshold$
   - Queue it
3. If $MinThreshold < AvgLen < MaxThreshold$
   - Calculate the *dropping probability* → P
   - Drop the packet with probability P
4. If $MaxThreshold \leq AvgLen$
   - Drop the packet

Note: Since $AvgLen$ changes slowly over time, the queues may be longer than $MaxThreshold$ at any given time.

– If there is really no room for the packet, (tail) drop it

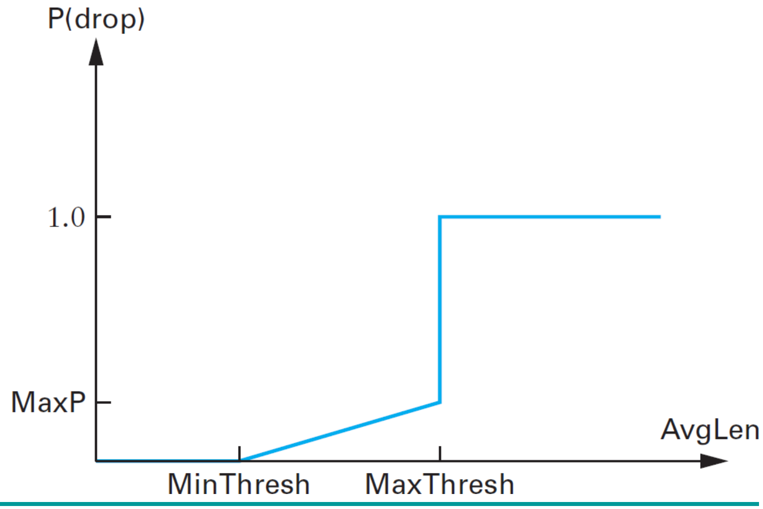
RED Drop Probability

The dropping probability changes over time and is affected by how long it's been since the last drop

- Prevents clustering of drops in bursty sending

Define $MaxP$ (maximum drop probability)

- $TempP = MaxP \times \frac{AvgLen - MinThreshold}{MaxThreshold - MinThreshold}$
- $P = \frac{TempP}{1 - (count \times TempP)}$



RED Dropping Example 1

$$MinThreshold = 10, MaxThreshold = 20, MaxP = 0.02, Count = 0$$

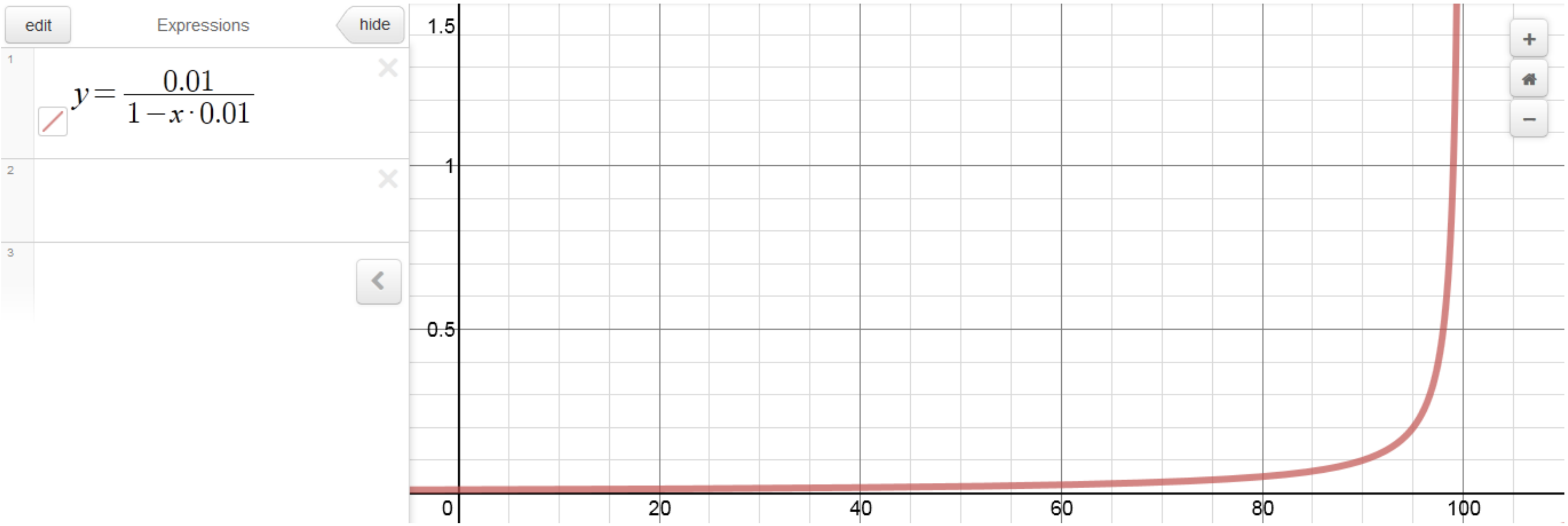
1. Assume $AvgLen = 15$

2. $TempP = 0.02 \times \frac{15-10}{20-10} = 0.01$
   $P = \frac{0.01}{1-0\times0.01} = 0.01$ ($\frac{99}{100}$ packets make it)

3. If $count = 50$ (no drops in 50 packets)
$$TempP = 0.02 \times \frac{15-10}{20-10} = 0.01$$
$$P = \frac{0.01}{1-50\times0.01} = 0.02$$

4. If $count = 99$ (no drops in 99 packets)
$$TempP = 0.01$$
   $P = \frac{0.01}{1-99\times0.01} = 1$ (the next packet is dropped)

RED Dropping Example 1



RED Dropping Example 2

$MinThreshold = 10, MaxThreshold = 20, MaxP = 0.02, Count = 0$

1. Assume $AvgLen = 19$

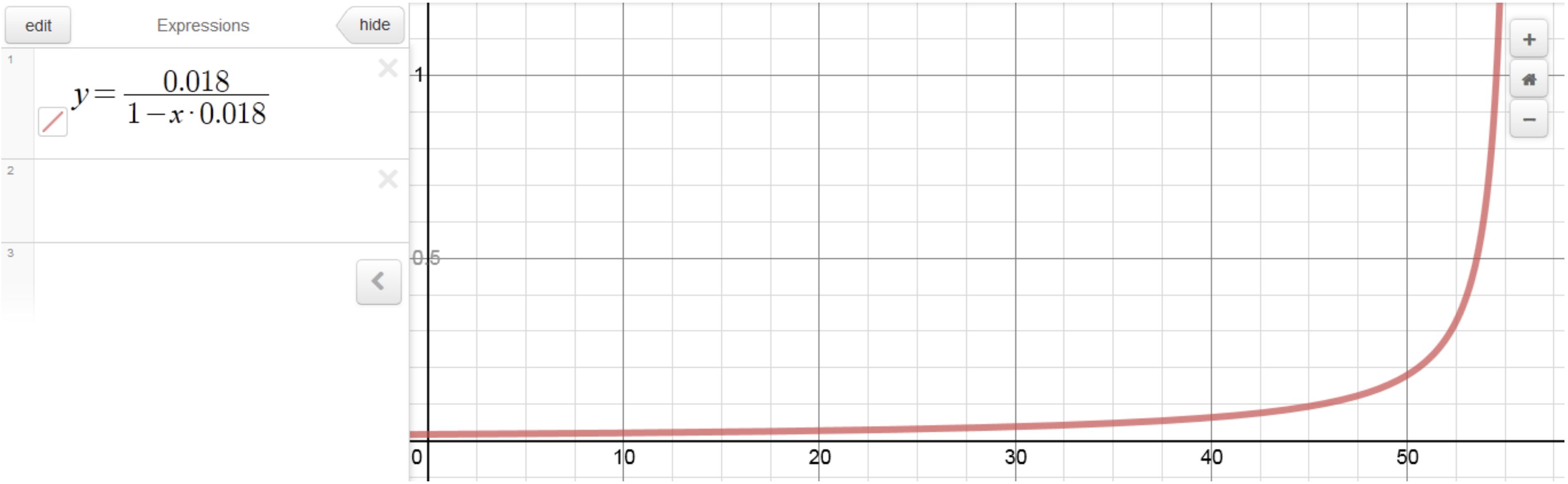
2. $TempP = 0.02 \times \frac{19-10}{20-10} = 0.018$
   $P = \frac{0.018}{1-0\times0.018} = 0.018$ ($\sim\frac{98}{100}$ packets make it)

3. If $count = 50$ (no drops in 50 packets)
   $$TempP = 0.02 \times \frac{19-10}{20-10} = 0.018$$
   $P = \frac{0.018}{1-50\times0.018} = 0.18$ ($\frac{82}{100}$ packets make it)

4. If $count = 54$ (no drops in 54 packets)
   $TempP = 0.018$
   $$P = \frac{0.018}{1-54\times0.018} = 0.6429$$

5. If $count = 55$ (no drops in 55 packets)
   $TempP = 0.018$
   $P = \frac{0.018}{1-55\times0.018} = 1.8$ (the next packet is dropped)

RED Dropping Example 2



Conclusion

- Fair Queuing
- Congestion Avoidance
  - RED