2.1 Non-work conserving scheduler

A non-work conserving scheduler can be idle even if there are packets that need to be served. The packets are only sent when they become eligible before that they are delayed. This implementation can help to make the downstream more predictable and eliminate the danger of bursts and make the traffic more smoothly and persistent. Because of this the buffer sizes and jitter can be reduced.
The downside of this implementation is that some packets can be stuck at such router and can delay for a significant amount of time, because the router does not recognise that packet as eligible and does not send it.

2.2 Work-conserving scheduler

2.2.1 Definition

A work-conserving scheduler is only idle when its packet queue is empty. A most common implementation of a work-conserving scheduler is a FCFS-scheduler. Furthermore the conservation law is beinng held at those type of schedulers.

2.2.1 Queue delay

Because we know from the lecture that:

$$\sum_i p_i \times q_i = \ const$$

we can formulate the following:

$$\frac{5}{150}*0.4+\frac{10}{150}*0.6+\frac{7}{150}*0.5+\frac{5}{150}*0.4 \ = \ \frac{5}{150}*0.3+\frac{10}{150}*0.7+\frac{7}{150}*0.4+\frac{5}{150}*x$$

so therefore we get the new delay for D:

$$x \ = \ \frac{11}{25} \ = \ 0.44$$

2.3 RED packet dropping scheme

t_0 : EAQL is between TH_{MIN} and TH_{MAX} $\Rightarrow$ Drop.Prob. > 0.5 $\Rightarrow$ is dropped
t_1 : EAQL is between TH_{MIN} and TH_{MAX} $\Rightarrow$ Drop.Prob. < 0.5 $\Rightarrow$ is queued
t_2 : EAQL is between TH_{MIN} and TH_{MAX} $\Rightarrow$ Drop.Prob. > 0.5 $\Rightarrow$ is dropped
t_3 : EAQL $> TH_{MAX}$ $\Rightarrow$ is dropped
t_4 : EAQL is between TH_{MIN} and TH_{MAX} $\Rightarrow$ Drop.Prob. < 0.5 $\Rightarrow$ is queued
t_5 : EAQL $< TH_{MIN}$ $\Rightarrow$ is queued
t_6 : EAQL is between TH_{MIN} and TH_{MAX} $\Rightarrow$ Drop.Prob. < 0.5 $\Rightarrow$ is queued
t_7 : EAQL is between TH_{MIN} and TH_{MAX} $\Rightarrow$ Drop.Prob. < 0.5 $\Rightarrow$ is queued

2.4 Advantages of early drop

An early drop is then an advantage when a router can predict that in the near future there will be a huge load on this link. Especially because a TCP connection needs a lot of time and the connection must be stable, it is better to drop it early than late which can lead to a loss of time which can be used to find a stable connection.

2.5 Different Types of schedulers

1. RR: A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, ...
   B would benefit from this implementation the most because it has only a few (only 2) packets, which need to be served. On the otherhand D is at a disadvantage because it has many, small packets.
2. DRR: C_1, D_1, D_2, A_1, C_2, D_3, D_4, A_2, A_3, B_1, C_3, D_5, D_6, B_2, C_4, D_7
   D would benefit from this implementation because its packets are relatively small and in many turns multiple packets of D are served. Further C would also benefit from it, because its packet size is roughly equal to the quantum size so it gets served in every round.
3. WFQ: A_1, B_1, C_1, D_1, D_2, C_2, A_2, D_3, D_4, D_5, C_3, A_3, B_2, D_6, C_4, D_7
   Because WFQ is an approximation of GPS it is relatively fair for all queues, so no queue is considerably benefitting. Some could argue that queue D is benefitting because its packages are short and follow each other in a relative short time.