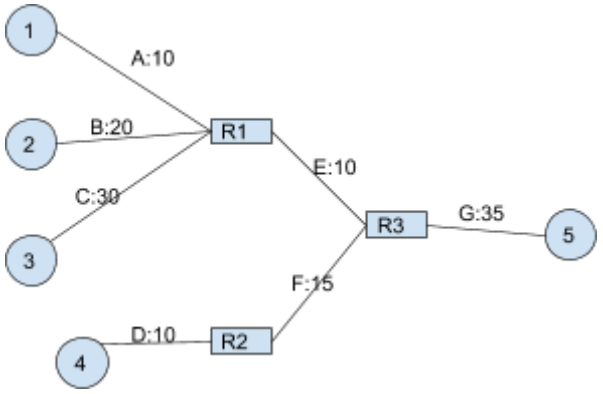
- **Traffic Management with Fair Queueing and Weighted Queueing**

Consider the network shown below. R1, R2, and R3 are routers and 1, 2, 3, 4 and 5 are hosts. The id and capacity (bandwidth) of each link (in Mbps) are labeled. Suppose 1, 2, 3 and 4 only send packets to 5 while 5 does not send any packets.



(a) **Based on the topology above, what is the maximum throughput (in Mbps) that can be achieved for the link G in the direction from R3 to host 5?**
20

(b) **If you are a network engineer and you want to choose a link and increase the bandwidth to fully utilize the link G, which link would you increase the bandwidth the first?**
Link E, it's the major bottleneck among R1 and R3 where there are a lot of hosts at the other end of R1 could get stuck due to the limit between R1 and R3. (As long as the student solution comes with a reasonable explanation, points would be awarded.)

(c) **Suppose R1 receives the following packets in the order listed at about the same time. Assume all incoming queues are empty and large enough to keep all receiving packets. (No dropping) The output port is busy.**

Packet	Size (in bytes)	Flow	Weight
1	300	A	1
2	150	B	2
3	120	B	2
4	200	C	4
5	150	C	4

i. Give the order in which these packets leave R1 if fair queueing is applied.
<2, 4, 3, 1, 5>

ii. Give the order in which these packets leave R1 if weighted fair queueing is applied.
<4, 2, 5, 3 ,1>

- **Traffic Management with Token Bucket**

Assume that we have a token bucket shaper.

1. **The token bucket size is 50KB and is full at time=0s.**
2. **r = 5 KB/s. Only replenish the bucket at the end of each second**
3. **Packets pass through the token bucket instantaneously if the token they need is sufficient.**

Also assume that a sender emits 10KB-packets every 0.5 seconds, starting at t=0.5s.

(a) How many tokens are left immediately after the third packet was sent? Show your work for partial credit.

At t = 0, left over = 50 KB.
At t = 0.5, left over = 50KB - 10KB = 40KB
At t = 1, left over = 40KB - 10KB + 5KB = 35KB
At t = 1.5, left over = 35KB -10KB = 25KB
At t = 2, left over = 25KB -10KB + 5KB = 20KB

(b) If we indexing the packets starting from 0, which packet would be the first to get queued due to the scarcity of token?

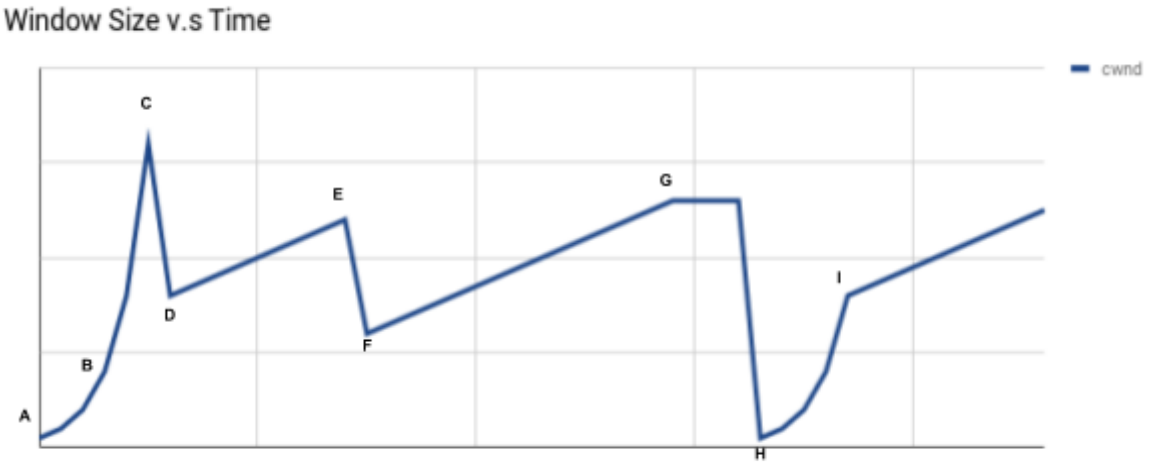
At t = 0, left over = 50 KB.
At t = 0.5, left over = 50KB - 10KB = 40KB id=0
At t = 1, left over = 40KB - 10KB + 5KB = 35KB id=1
At t = 1.5, left over = 35KB -10KB = 25KB id=2
At t = 2, left over = 25KB -10KB + 5KB = 20KB id=3
At t = 2.5, left over = 20KB -10KB = 10KB id =4
At t = 3, left over = 10KB -10KB+5KB = 5KB id =5, still got sent
At t = 3.5, packet with id as 6 would not have enough token to get sent.

Thus, the answer is a packet that has id as 6.

(c) Now, we turn the infinite maximum rate R to 15 KBps, and we refill the bucket continually with r = 5 KB/s, (instead of refill at the end of each second), what would the maximum burst size be?

We use the formula: $b * R/(R - r) = (50KB * 15KBps)/(15KBps - 5KBps) = 75KB$.

- **TCP Congestion Control**



(a) Fill in the blanks in this table according to the diagram above:

	A	B	C	D	E	F	G	H	I
Time (in the unit of RTT)	0	3	5	6	14	15	29	33	37
CWND	1	8	32	16	24	12	26	1	16

(b) Based on the diagram and table, make some Inferences.

1. How would you describe the stage between A and C?

   TCP Slow Start
2. How would you describe the stage between D and E?

   AIMD/Congestion Avoidance
3. How would you describe the stage between F and G?

   AIMD/Congestion Avoidance
4. How would you describe the stage between H and I?

   Slow Start
5. What would ssthresh be after the timeout at H?

   13

- **Modulation**

Draw the waveforms of the input signal, and the amplitude shift, frequency shift, and phase shift keying of the input for the input bit sequence that's provided at the top. The waveforms have been provided for the first two bits of the sequence, to help you get started.

You were not required to draw perfect waveforms, especially for PSK. But do try to present the techniques they used to represent different bit information and transition.