HW3

1. 1. A minimum packet size is equivalent to the slot time, or the time it takes for one packet to transverse the diameter of the network in a round trip time. This means the sender has a lower bound of time to see whether a collision occurred or not.
   2. I need to find the time required to make the minimum packet size or slot time. Given speed and distance, go back to physics to get:

v = d/t

2\*108 m/s = 500m/t

500m / 2\*108 m/s = t

t = 2.5 \*10-6 seconds

Then we use the given rate, the time derived, and x2 for a RTT:

100 Mbps \* 2.5 \*10-6 seconds \* 2 = 500 bits

So **500 bits** minimum packet length. This will be larger in Ethernet since you need some bits for a jamming signal.

2. S would send packet to AS 100 since it is the only link available. AS 100 would then run a check to see the shortest way to D using inter and intra domain algorithms, Distance Vector and Link State.AS 100 will see that the response time from AS 200 to interior router R4 is faster than routing iBGP through R2 and R3 and thus send the packet to AS 200 R5. AS 200 will use interdomain eBGP to send the packet back through AS 100where the packet is received by gateway router R4. R4 will then send the packet to D.



* 2. Min threshold is determined by the utilization requirement. minth must not be too low that it drops all incoming packets but not too high that it drops no packets and leads to congestion.

Max threshold manages the tradeoff between acceptable maximum queuing delay for packets such that a network can get out of a congested state. maxth is used when the network has a lot of packets coming in. If it is too large then no packets dropped means sync will occur and congestion will be unavoidable, but if it is too small then many packets will be dropped and flows will have larger delays.

* 1. Linear increase means that it will be highly possible two flows will be synced since the RED router will mark multiple flows with the same probability.

1. 1. For max-min fairness, we give evenly give each user a small portion of resources that they demand for and any increase to a user will decrease another user that has equal number of resources.

9 units / 5 demands = 1.8 w/ .8 excess for A

.8 / 4 demands = .2 => [1, 1.2, 1.2, 1.2, 1.2] allocated, B has excess of .2

.2 / 2 = .1 => **[1, 1, 1.4, 1.4, 1.4]** is the final allocated values to A,B, C, D, and E.

1. Fair queueing is the act of dropping packets from different flows equally in network-congested scenarios. If a certain flow wants to gain bandwidth they can just flood the switch to occupy the buffer all the time. That way all other flows’ packets will be dropped. Fair Queuing identifies misbehaving flows by not reporting back explicitly to the source the misbehaviors are being dropped by the switch and by keeping track of misbehaving flows.
2. 1. Since this would keep track of every packet in the queue, it’d be complex. However, the dropped packets are using too many resources that are good to drop if they remain in the queue too long.
   2. This can happen with 3 packets where the sizes and order are 3, 7, 2. Since we calculate the size x sum of sizes of previous packets, the last packet of size 2 has a cost of 22 while the middle larger packet of 7 has a cost of 21.
3. Instead of the core router maintaining per-flow state control, edge routers maintain the state and mark packets with a Dynamic Packet State. Core routers then can just see the aggregate flows, maintain a fair rate share value, and drop packets based on this fair share value. In comparison to FQ, this reduces the complexity of core routers and results in throughput increases. However, edge routers will need more complexity and configuration. XCP manages aggregate flows via a control interval and uses a fairness and congestion controller.