# CS 350 – Fall 2020, Assignment 3

**Answer 1**

Since at maximum clock speed (1GHz) the system can handle 200 Requests per second, then it means it finished 0.2 requests per ms, or it takes 5ms to complete a single request.

a) If we do the same math on customer A's request, we get that they only need to have a request fulfilled every 16.6 second, which gives the system more than enough time to guarantee QoS.

b) Despite customer B only asking for a new request every 33.3 second, and since the latency required is 5ms, and that is as fast as the system works, we can also guarantee QoS.

c) Finally, by the same logic, we can also guarantee the QoS for customer C.

d) For Customer A I can slow down the CPU by two thirds to 0.3GHz. For Customer B I can slow down to 0.15GHz. For Customer C I can only slow down by to 0.25GHz, as the latency allows for it.

e) For Customer A there will be a queue of either 0 or 1, since the system will handle any requests before another one arrives (on average). For Customer B the same thing will occur. For Customer C however, there will be an average queue of 1-2 requests

f) Since utilization is exactly 1 for Customer A, then the probability of it finding an element there is 0. For Customer B the utilization is also 1, so the probability is also 0. Finally for Customer C probability is 100%, as we are making use of the fact that the customer allows for further latency to serve lower speeds.

**Answer 2**

a) Since the switch we selected can handle $10^8$ bits, and the system is going to handle an average of 8200 packets of 1.2KB (or 1228.8 bytes, or 9830.4 bits each). Then the router has extra bandwidth, since it can handle $10^8$ bits, meaning it can handle 10172.5 requests per second. Which means we'd need an overflow of 1972 requests before the system overloaded. With that we can problematically assume that the queue for the router is either 0 or 1, meaning that the router will need a memory of 2.4KB or 2457.6 Bytes.

b) Since we discussed in the previous answer that the queue will either be 0 or 1, if we calculate the probability of each of these events we get P(0) = 0.16, and P(1) = 0.128. Since we know that a single request takes 0.98ms, then we can use a weighted average to get that the average timestamp difference will be 0.125 ms.

c) This means that if the utilization is above 70% we'll need to cool the system. We can calculate the utilization of the system as $utilization = averageArrivalRate \times averageServiceTime = 8200 \times 0.00098 = 0.80$ or 80% utilization, so we'll need to cool the system.

d) I assumed that the system is in steady state and didn't leave steady state, I also assumed that my answers to previous questions were always right when answering the next.

e) If we compare the 100Mbps speed to the 1Gbps speed, we have a speedup of $10^9/10^8 = 10$, or 10x faster than the previous router.

f) Due to the increased overhead, if the service requirements are maintained, then we only really need enough memory to hold the request that is currently being processed, or 1.2KB (1228.8 Bytes), cutting our memory requirements by half.

**Answer 3**

a) Since we know we have an average queue length of 20, and each request is taking 35 seconds to complete, that means that, from the moment it arrives it'll be on average the 20th request in the queue, which means the response time will be 700 seconds, which means the system will be handling 0.028 Requests a second.

b) If we take our theoretical capacity to be the average rate of arrival with 100% utilization, then we have that: $1.0 = arrivalRate \times serviceTime \equiv \frac{1.0}{serviceTime} = arrivalRate$, since we know it takes each request 1.75 seconds to be completed we have that: $\frac{1.0}{35} = 0.028$. This means that the system will bottleneck if requests arrive any faster than an arrival rate of 0.028 per second.

c) No, since the system is already bottlenecking as is.

d) If we re-do the calculations above for the capacity we get that the maximum arrival rate with a service time of 25 is 0.04. Meaning we could increase the traffic 42.8% before bottlenecking.

e) If we improve service time by 20%, then our service time will be 28 seconds, which means we could service 0.035 requests per second, or an increase of 25%, however, since we are cutting our prices by half, we'll see a decrease of 37.5% when compared to the initial profit.