# ECE 592 005 – IOT Analytics

# Project 1 – Simulation Task 3

The values used to plot the graphs are in the Excel sheet attached below:



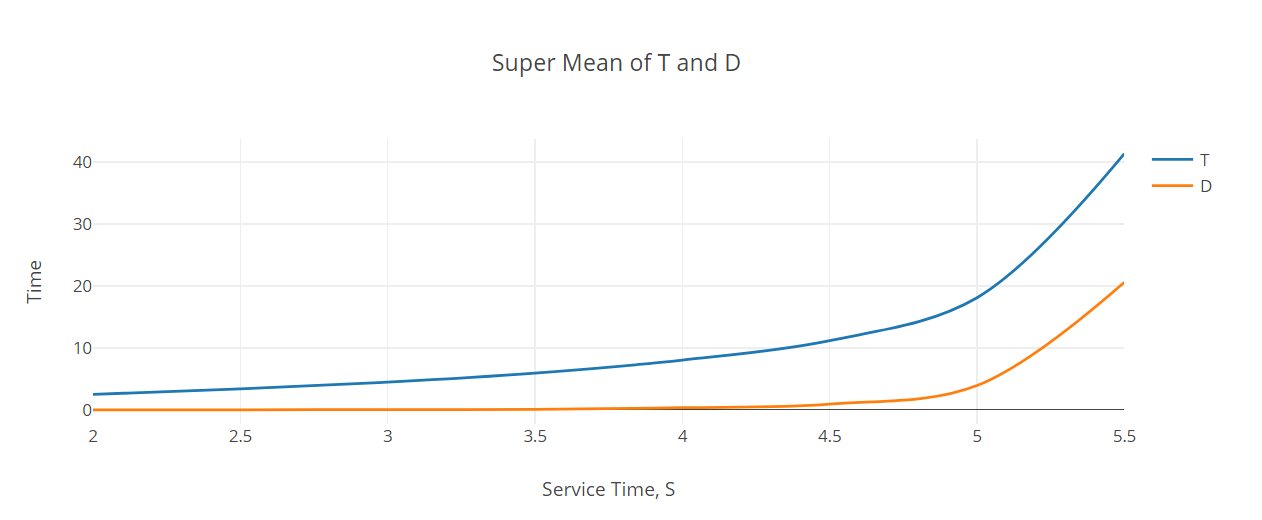
Task-1:

The Mean , 95th percentile and Confidence Intervals for various Service times, S = 2,2.5,3,3.5,4,4.5,5,5.5 were calculated and the results were plotted on a graph.

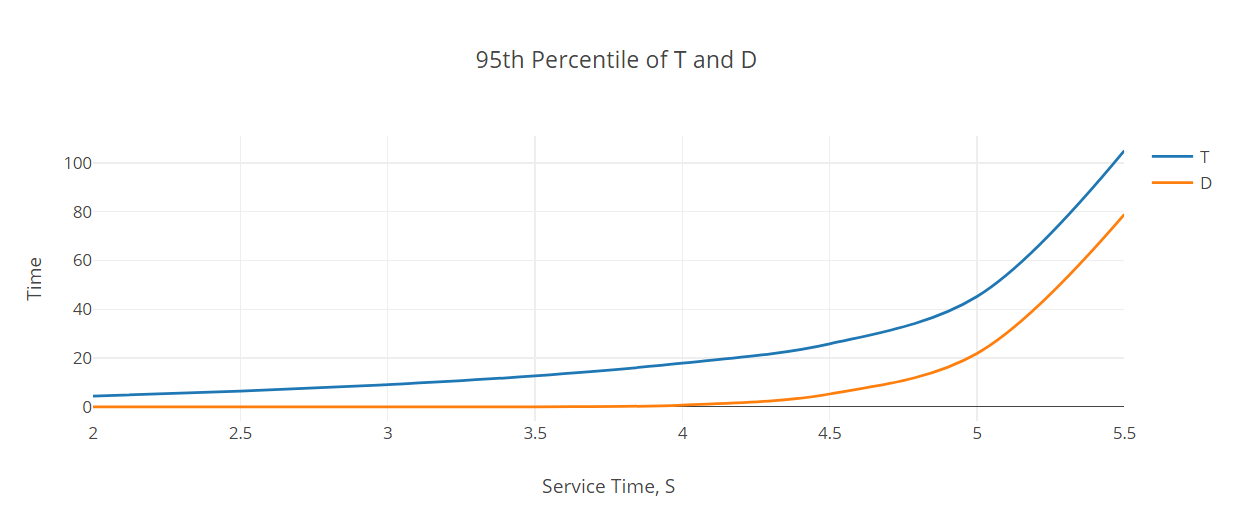
Buffer B = 5

Seed = 5

1. Plot of the T and D mean values VS Varying Service Times S



1. Plot of the T and D 95th Percentile values VS Varying Service Times S



Observations and Inferences out of Task-1:

* As the service time is increased, the following values increase exponentially.
  + Mean Time taken for Recertification (T)
  + Mean Retransmission Time (D)
  + Mean of 95th Percentile of the total time for Recertification (T95th)
  + Mean of 95th Percentile of the Retransmission time (D95th)

This can be seen in the graphs plotted above. We get a curve that increases exponentially as S value increases.

* This is because we only consider a single server that is accessed for recertification, which handles requests in a linear fashion and only one request can be serviced by the server for recertification. Hence, as the service time increases, the buffer gets full and we see further new requests being sent for retransmission directly.
* As the T and D mean / percentile values increase exponentially, we also note that when comparing the T and D mean for higher service time values, the values are almost equal. This is because new requests are directly sent to retransmission and then later joins the buffer when a running service completes.
* We also observe an increase in the confidence intervals for both the T and D means and the T and D 95th Percentile values increase with the increase in service times. This is undesirable because confidence intervals are values that help in measuring values accurately. A larger confidence interval implies a less accurate runtime value for the recertification process.
* From the above observations, we see that a larger service time leads to greater number of issues in the system. Also, using just a single server that services requests in a linear fashion is also a downside. The following changes if done can improve the efficiency of the system:
  + Maintain a low service time and a moderate Buffer size
  + Making the single server service parallelly and certify several servers at once through multi-threads/multi-processes or increasing the number of servers in the system.

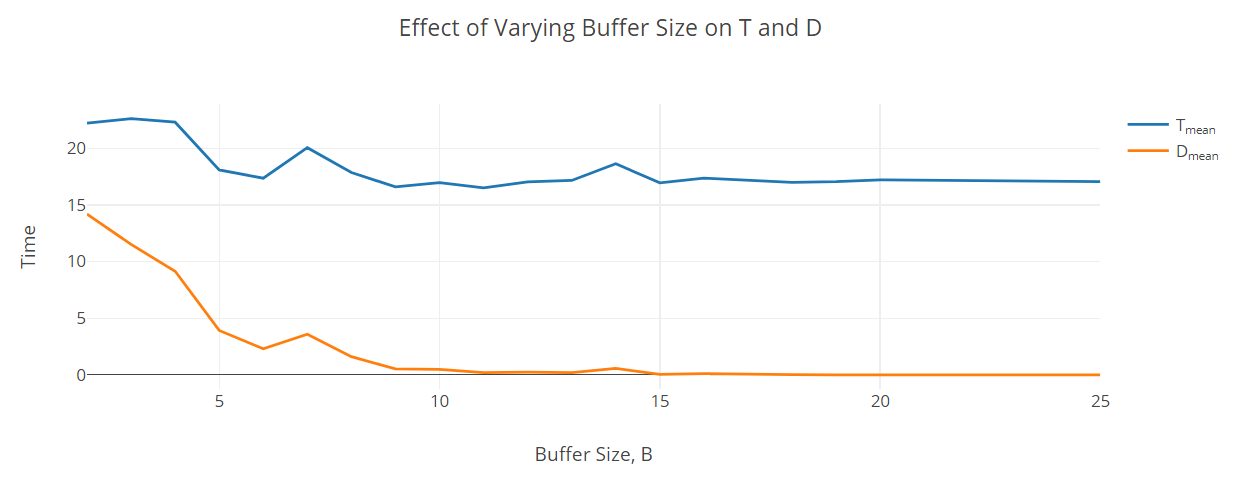
Task-2:

The Mean , 95th percentile and Confidence Intervals for various Buffer Sizes B = 2 to 20 and 25 were calculated and the results were plotted on a graph.

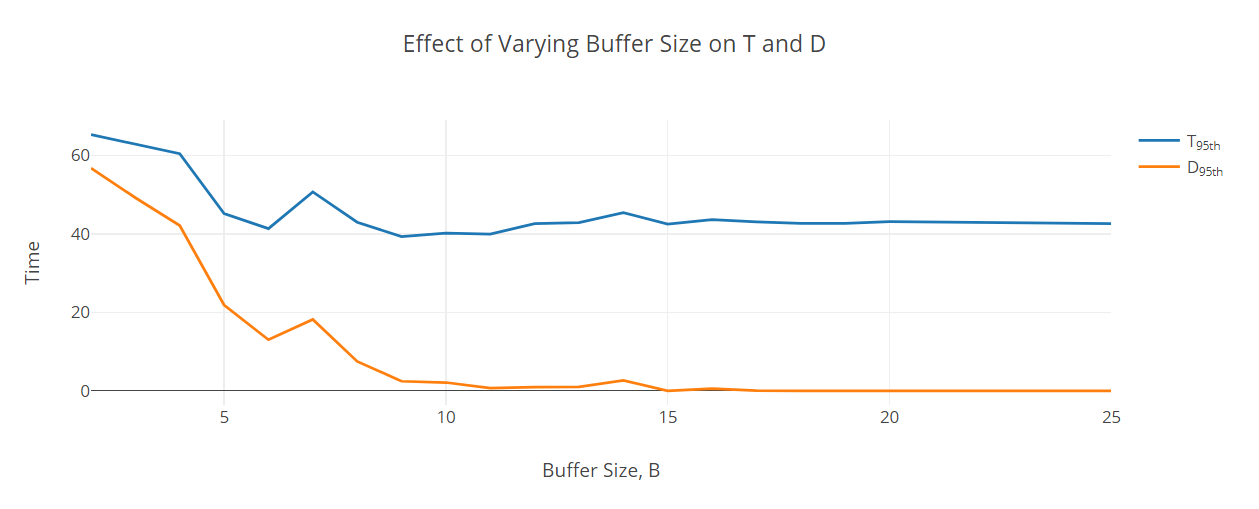
Service Time S = 5

Seed = 5

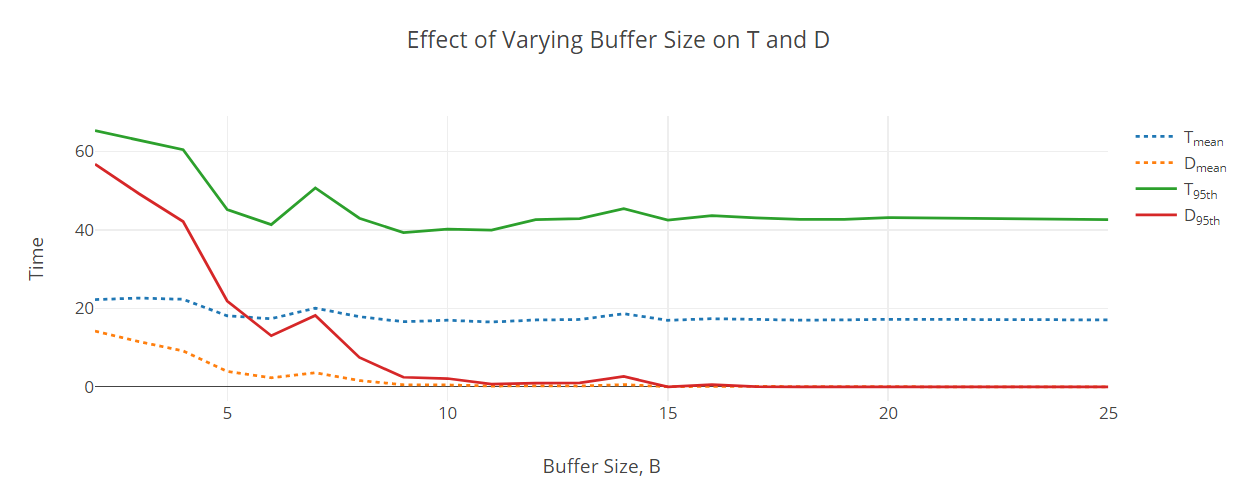
1. Plot of the T and D mean values VS Varying Buffer Size B



1. Plot of the T and D 9th Percentile values VS Varying Buffer Size B



1. Consolidated Plot of Tmean , Dmean , T95th , D95th values VS Varying Buffer Size B



Observations and Inferences from Task-2:

* As the Buffer Size is increased, the mean values decrease gradually.
  + The following mean values decrease and become zero after a few increases in the buffer size,
    - Mean Retransmission Time (D)
    - Mean of 95th Percentile of the Retransmission time (D95th)
  + The following mean values decrease and stabilize after a few increases in the buffer size,
    - Mean Time taken for Recertification (T)
    - Mean of 95th Percentile of the total time for Recertification (T95th)
* This is because as the buffer size keeps increasing, the system has more room for requests to be accommodated. The system sort of attains a state of equilibrium achieving almost same mean and 95th percentile recertification time as buffer size does not matter anymore. The mean and 95th percentile retransmission time also becomes 0 for higher buffer values as the new requests get serviced almost immediately as they enter into the system.
* The Retransmission time D turns out to be 0 for almost all values above 17, with a service time of 5 and seed value 5.
* A buffer size of 1 would have made the system similar to a Stop and Wait but without the acknowledgement. The server would service only one request, and the system gets blocked until it is serviced. No other new requests can enter until service completion time. This type of behavior is not expected from such a recertification system.