Math 523 Project 3

Xi Sun

July 23, 2019

1 Introduction

In this project, we are going to simulating a queuing system and trying to compare the theoretical throughput X, utilization U, and $\mathrm{E}(N)$ with the result we get from this simulation.

2 Modeling and Analysis

In the real world application, the parameters usually being found from historical data. But in this project, we will assign the parameter for arrival process $\lambda = 1$ and the parameter for service time $\mu = 2$.

```
queuing <- function(lambda,mu,N){
  arrival <- rexp(N,rate=lambda)
  cumarr <- cumsum(arrival)
  service <- rexp(N,rate=mu)
  cumser <- cumsum(service)
  idle <- c()
  idle[1] <- arrival[1]
  n <- length(arrival)
  for (i in 1:(n-1)){
    idlesum <- sum(idle)
    if (idlesum+cumser[i]<cumarr[i+1]){
      idle[i+1] <- cumarr[i+1]-(idlesum+cumser[i])
    }else{
      idle[i+1] <- 0
    }
  }
  sercomp <- cumarr+service
  cumidle <- cumsum(idle)
  df <- data.frame(arrival_time=cumarr,service_time=service,
                   service_completion=sercomp,idle,cumulative_idle=cumidle)
  return(df)
}
queuing(lambda=1,mu=2,N=10)

##    arrival_time service_time service_completion      idle cumulative_idle
## 1      3.147207    1.5031675           4.650375 3.1472070        3.147207
## 2      3.706651    0.2308192           3.937470 0.0000000        3.147207
## 3      4.520623    0.1795370           4.700160 0.0000000        3.147207
## 4      6.296546    0.6367579           6.933304 1.2358157        4.383023
## 5      6.415383    0.2771331           6.692516 0.0000000        4.383023
## 6      6.677727    1.4673705           8.145097 0.0000000        4.383023
## 7      7.249326    0.3048098           7.554136 0.0000000        4.383023
## 8      9.426259    0.3675164           9.793776 0.4436415        4.826664
## 9     11.428924    0.6161068          12.045031 1.6351483        6.461813
## 10    15.384608    1.2173045          16.601912 3.3395772        9.801390
```

The above dataframe gives a visulization of how the arrival, service and idle time being set, now we need to check the values.

```
queuing <- function(lambda,mu,N,tmax){
  arrival <- rexp(N,rate=lambda)
  cumarr <- cumsum(arrival)
  x <- sum(cumarr<=tmax)/cumarr[sum(cumarr<=tmax)]
  u <- x/mu
```

```
  return(c("throughput"=x,"utilization"=u))
}
queuing(lambda=1,mu=2,N=1000,tmax=1000)

##  throughput utilization
##   1.0176139   0.5088069
```

By taking maximum number of arrival $N = 1000$ and ending time tmax $= 1000$, we can see that the result for throughput and utilization is close to $\lambda = 1$ and $\rho = \frac{\lambda}{\mu} = \frac{1}{2}$. So the theoretical values are being checked in our simulation.

```
E_N <- function(lambda,N,ite,tmax){
  nvec <- c()
  for (i in 1:ite){
    arrival <- rexp(N,rate=lambda)
    cumarr <- cumsum(arrival)
    nvec[i] <- sum(cumarr<=tmax)/ite
  }
  en <- mean(nvec)
  return(en)
}
E_N(lambda=1,N=1000,ite=1000,tmax=1000)

## [1] 0.987609
```

According to the value for $\mathrm{E}(N)$ above, we can see that the values are close to 1, and in theory, $\mathrm{E}(N) = \frac{\rho}{1-\rho} = \frac{\frac{1}{2}}{1-\frac{1}{2}} = 1$. Therefore, $\mathrm{E}(N)$ in the simulation satisfies the theoretical value.

3 Conclusion

According to all the result from above, we can see that all the actual values matches the theoretical values. However, we have another limitation that this has to be in steady-state to make those values make sence, so we can conclude that as long as t is large, then we can actually apply those theoretical values in the real world applications.