SMDE_assignment03

Asaf Badouh, Pau Rodriguez

December 11, 2017

getMu - explained in the lab assignment paper.

Allen-Cuneen approximation's formula for G/G/1

base on miri-32-eng.pdf slide 7

```
getMu <- function(rho, E_tau, sigma){
  m = (E_tau * rho)/(sqrt(exp(sigma*sigma)))
  mu = log(m)
  return(mu)
}

C_s_theta <- function (lamda, mu, rho){
  numerator = (lamda/mu)/(1-rho)
  denominator = 1 + numerator
  return(numerator/denominator)
}

#C(s=1, theta)
allenCuneen <- function (lamda, sigma_tau, mu, sigma_x, rho) {
  C_theta = C_s_theta(lamda, mu, rho)
  #x to the power of y -> x**y
  numerator = lamda**2*sigma_tau**2 + mu**2*sigma_x**2
  numerator = numerator * C_theta
  denominator = 2 * mu * (1-rho)

  return(numerator/denominator)
}
```

```
runningQueue <- function(mu, sigma, clients=1000, myseed = -1) {


if (myseed > -1){
  set.seed(myseed)
}
service_time = rlnorm(clients,mu,sigma)
inter_arrival = rnorm(clients,77,15) #(t_i) - entrance time instant to the W.S.; it can be obtained thr
L = 0
Lq = 0
W = 0
Wq = 0
LT_i = array(0, clients)
L_i   = array(0, clients)
W_i= array(0, clients)
L_q = array(0, clients)
W_q = array(0, clients)
```

```
t_i = array(0, clients)
ts_i = array(0, clients)    #(t_i)^S= arrival time instant to the service system
theta_i = array(0, clients) #Theta_i - exit time instant from W.S. . for client i
for (c in 1:clients){
  #step #1
  if(c == 1)
    ts_i[c] = max(t_i[c])
  else
    ts_i[c] = max(theta_i[c-1], t_i[c])
  #step #2 - done in the pre-processing phase "service_time"
  #step #3
  theta_i[c] = ts_i[c] + service_time[c]
  #step #4
  if(c < clients)
    t_i[c+1] = t_i[c] + inter_arrival[c]
  #step 5 : calculation and printing
  #5a
  L_i[c] = W_i[c] = theta_i[c] - t_i[c]
  L = L + L_i[c]
  #TODO: check what about LT_i[1]
  if (c > 1)
    LT_i[c] = L/(t_i[c]-t_i[1])
  W = W + W_i[c]

  #5b
  L_q[c] = W_q[c] = ts_i[c] - t_i[c]
  Lq = Lq + L_q[c]
  Wq = Wq + W_q[c]

}

W = W/clients
Wq = Wq/clients
L = L/(t_i[clients] - t_i[1])
Lq = Lq/(t_i[clients] - t_i[1])

 # plot(t_i, LT_i, type="l", xlab="t_i", ylab="LT_i")
  res = list("t_i" = t_i, "LT_i" = LT_i, "W" = W, "Wq"= Wq, "L" = L, "Lq" = Lq)
  res
}
```

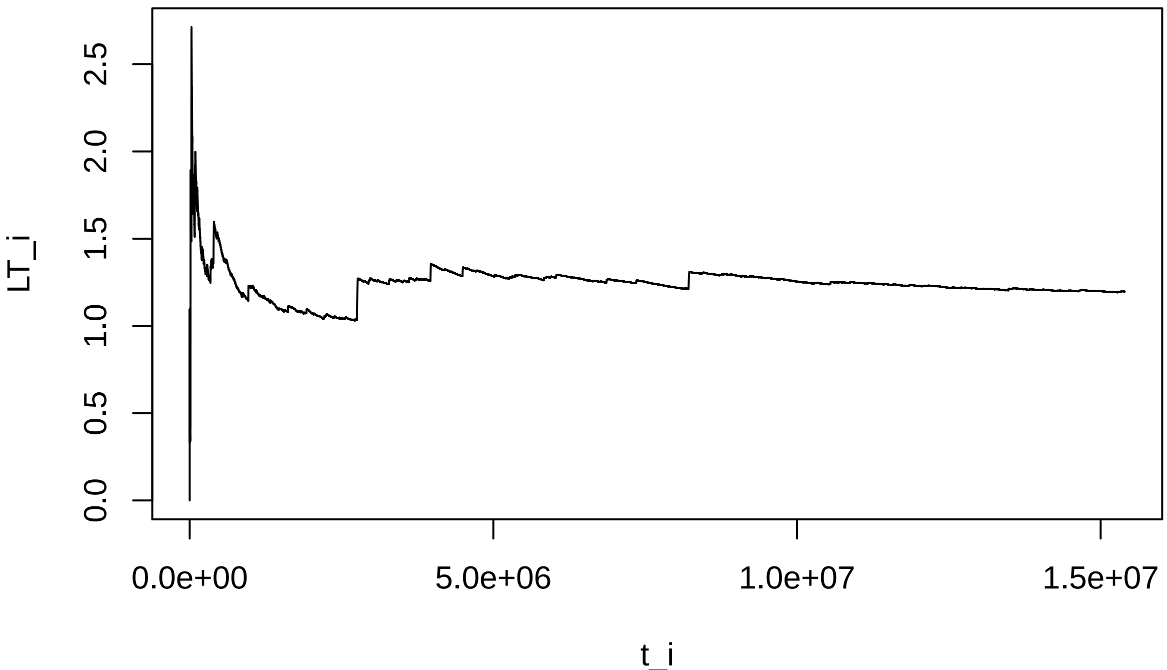
```
get_X_mu <- function(rho, X_sigma, Tau_mu, Tau_sigma) {

  x_mu = log(rho*Tau_mu/exp((X_sigma**2)/2))

  return(x_mu)

}

part1_2 <- function(rho, X_sigma, Tau_mu, Tau_sigma){

    x_mu = get_X_mu(rho, X_sigma, Tau_mu, Tau_sigma)

    C_x = sqrt(exp(X_sigma**2) - 1)
```

```r
    C_tau = Tau_sigma/Tau_mu

    Lq_mm1 = (rho**2)/(1 - rho)

    Lq_AllenCuneen = Lq_mm1 * (( C_x**2 + C_tau**2 ) /2)

    lambda = 1/Tau_mu

    Wq_AllenCuneen = Lq_AllenCuneen/lambda

    return( c(Lq_AllenCuneen, Wq_AllenCuneen, x_mu))

}
```

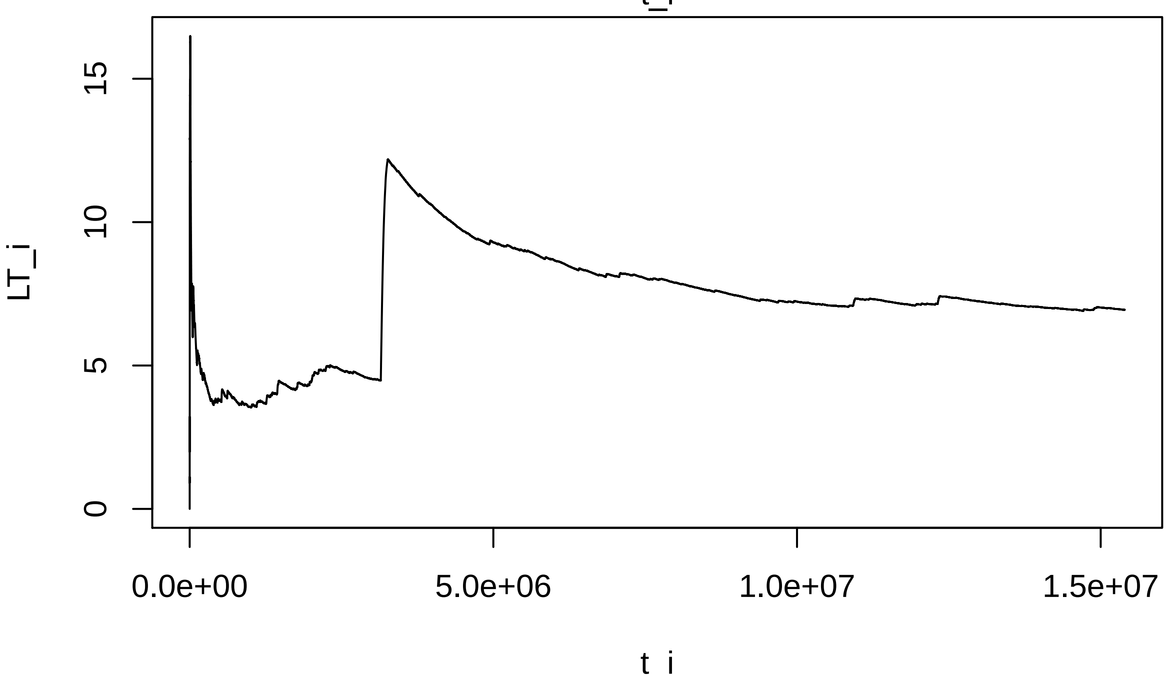
```r
clients = 200000
myseeds = c(7,13,109,211,273,711,777,1001,7001,99)
stat = list()
iterations = 5
for (p in 1:length(rho)){
  mu = getMu(rho[p], E_tau, sigma)
  print(mu)
  for(i in 1:iterations){
    stat = list.append(stat, runningQueue(mu, sigma,clients, myseeds[p %% 10 + 1]))
  }
}
```

```
## [1] 2.407066
## [1] 2.966681
## [1] 3.160838
## [1] 3.245395
```

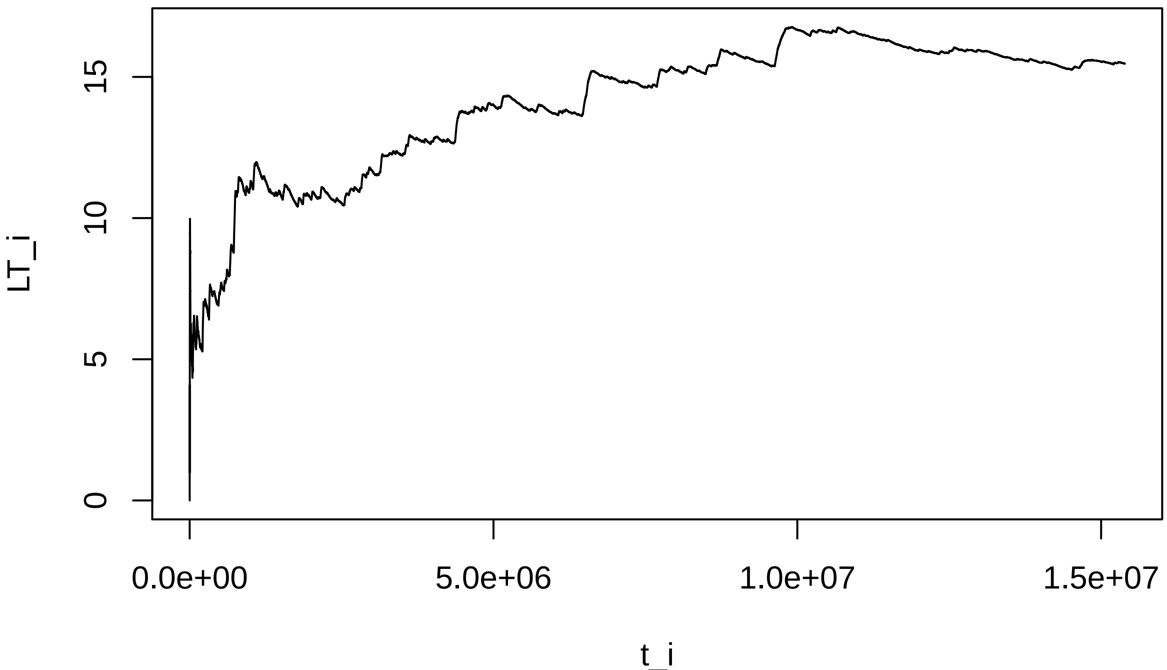
```r
# stat contain all the return values from running Queue where the indexing is:
# stat[1-iterations] results for rho[1] and the corrisponding mu
# stat[iteration+1 - 2*iteration] results for rho[2] and the corrisponding mu
# stat[2*iteration+1 - 3*iteration] results for rho[3] and the corrisponding mu
# stat[3*iteration+1 - 4*iteration] results for rho[4] and the corrisponding mu

for (p in 1:length(rho)){
  for(i in 1:iterations){
    #need to do better plotting,
    #maybe to plot all the iterations at with the same rho and mu on a same graph.
    plot(stat[i+iterations*(p-1)][[1]]$t_i, stat[i+iterations*(p-1)][[1]]$LT_i, type="l", xlab="t_i", yl
  }
}
```

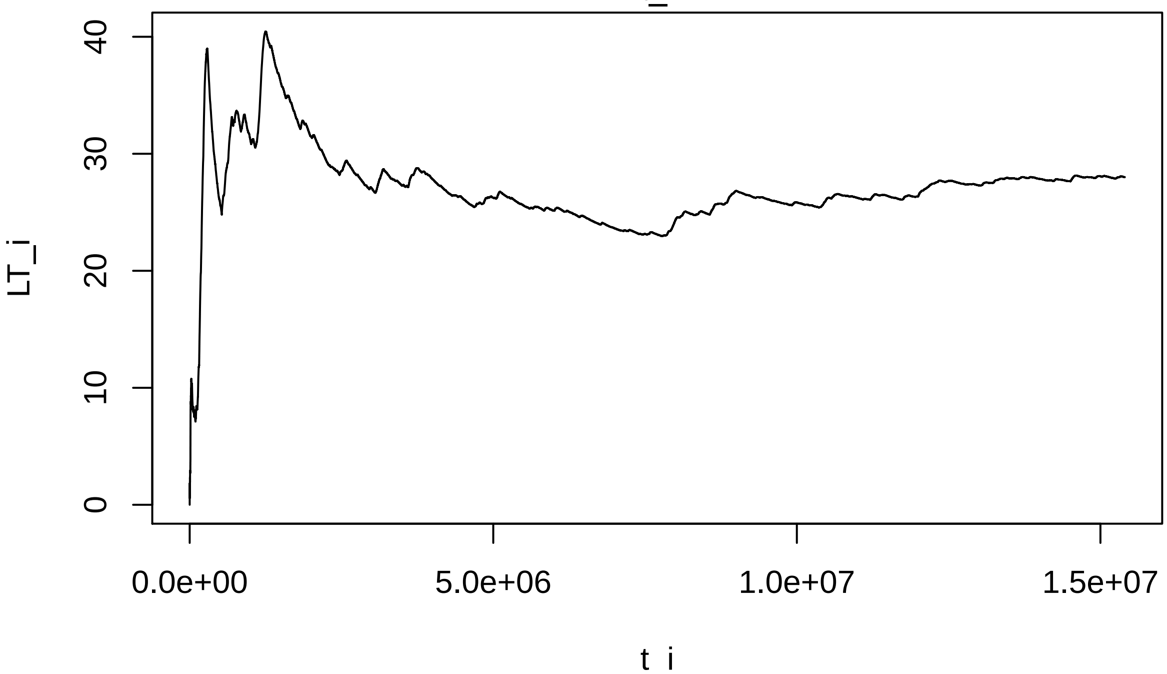


LT_i
0.0
0.5
1.0
1.5
2.0
2.5
0.0e+00
5.0e+06
1.0e+07
1.5e+07
t_i



LT_i
0
5
10
15
0.0e+00
5.0e+06
1.0e+07
1.5e+07
t_i





Initial analysis

```r
rho07 = read.table('./part1/rho_0_7/queue-sim-arribades.txt')
colnames(rho07)
```

```
## [1] "V1"
```

```r
rho07$V1[3]
```

```
## [1] 62.13841
```

```
# incomplete!
```

Allen Cuneen's aproximation formula for W_q and L_q

For each loading factor ρ, we derive the required μ value for the Lognormal distribution:

$$s = 1\lambda = \frac{1}{E[\tau]}\mu = \frac{1}{E[x]}E[x] = m \cdot e^{\frac{\sigma^2}{2}} = e^{\mu+\frac{\sigma^2}{2}}\rho = \frac{\lambda}{s\mu} = \frac{E[x]}{E[\tau]} = e^{\mu} \cdot \frac{e^{\frac{\sigma^2}{2}}}{E[x]} \Rightarrow \mu = ln\left(\frac{\rho}{e^{\frac{\sigma^2}{2}}} \cdot E[\tau]\right)$$

We use the Allen Cuneen's aproximation formula for L_q:

$$L_q \approx L_{q_{M/M/1}} \cdot \left(\frac{C_\tau^2 + C_x^2}{2}\right)$$

With:

$$C_x = \sqrt{\omega - 1} = \sqrt{e^{\sigma^2} - 1}C_\tau = \frac{\sigma_\tau}{E[\tau]}$$

And derive W_q:

$$W_q = \frac{L_q}{\lambda}$$

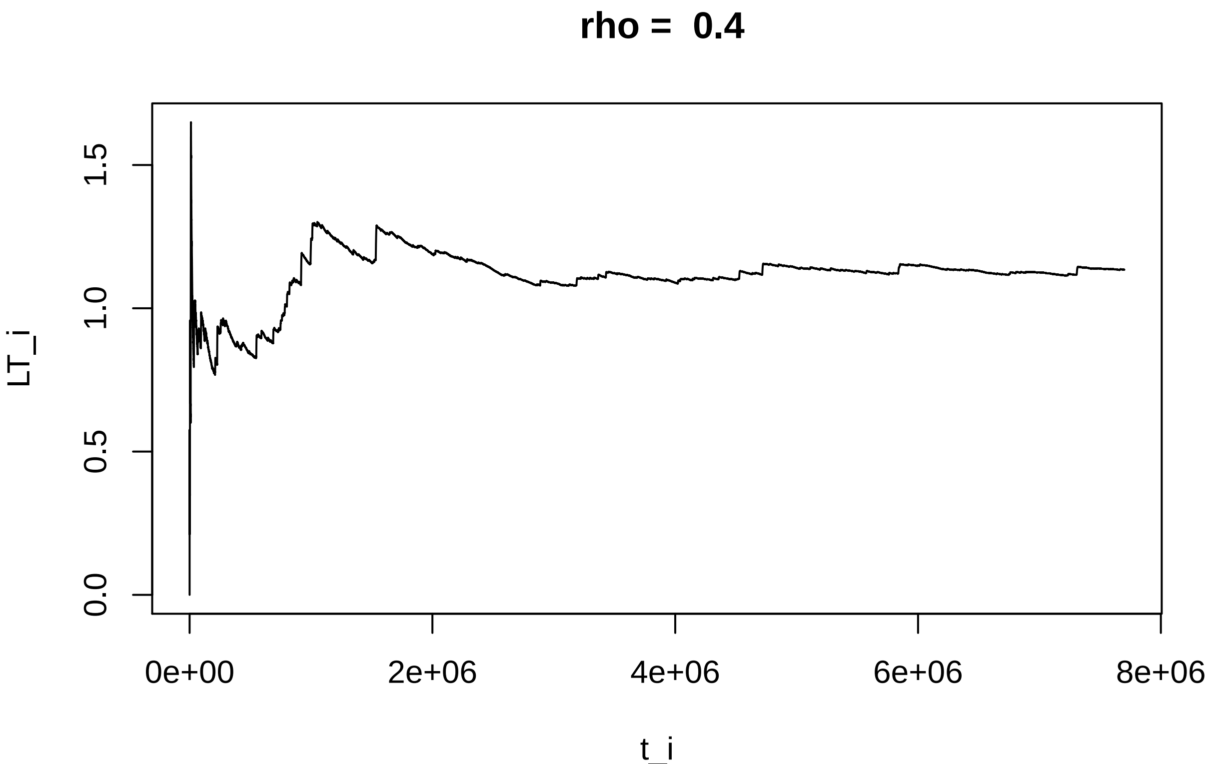
```
# part 1.2 Allen Cuneen aproximation calculation
aproximations = rep(c(0,0), 4)
i = 1
for (r in rho) {

  Lq_and_Wq_and_mu = part1_2(r, sigma, E_tau, E_sigma)

  aproximations[i] = Lq_and_Wq_and_mu[1]
  aproximations[i+1] = Lq_and_Wq_and_mu[2]
  aproximations[i+2] = Lq_and_Wq_and_mu[3]
  i = i + 3
}
```

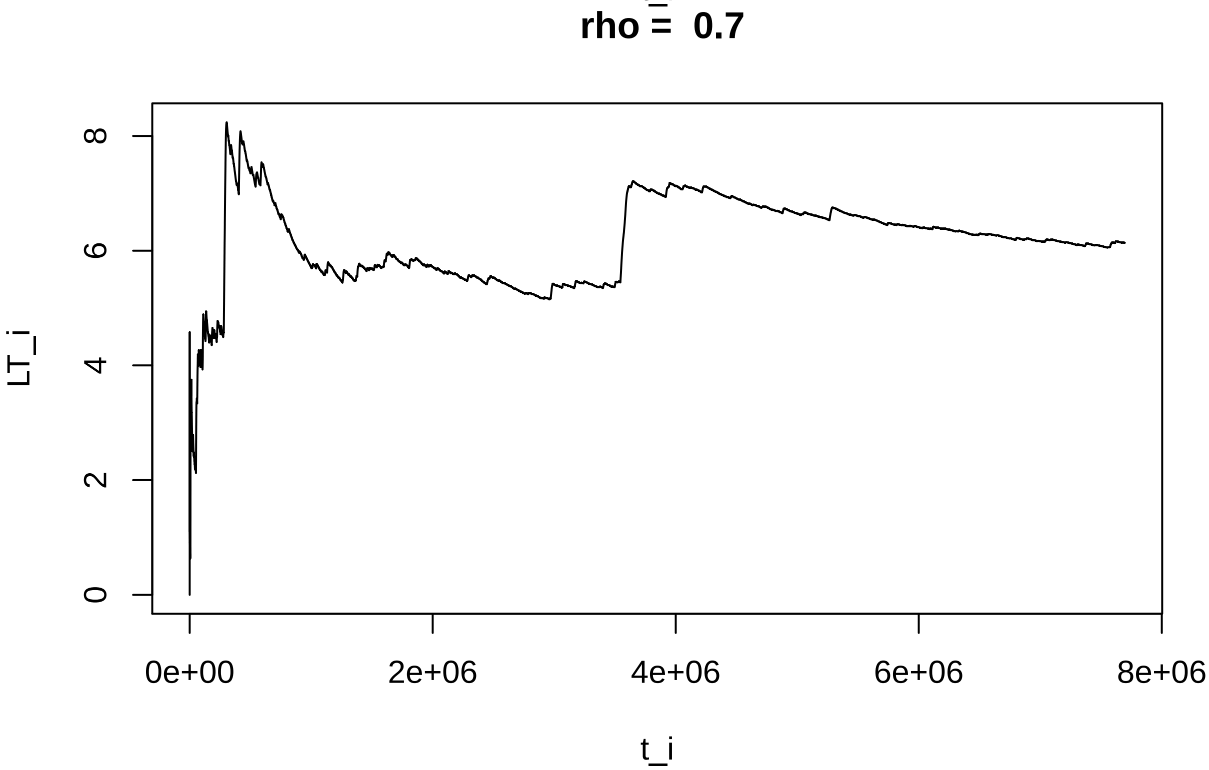
Using the Allen Cuneen's aproximation formula, we can compute the W_q and L_q for each loading factor:

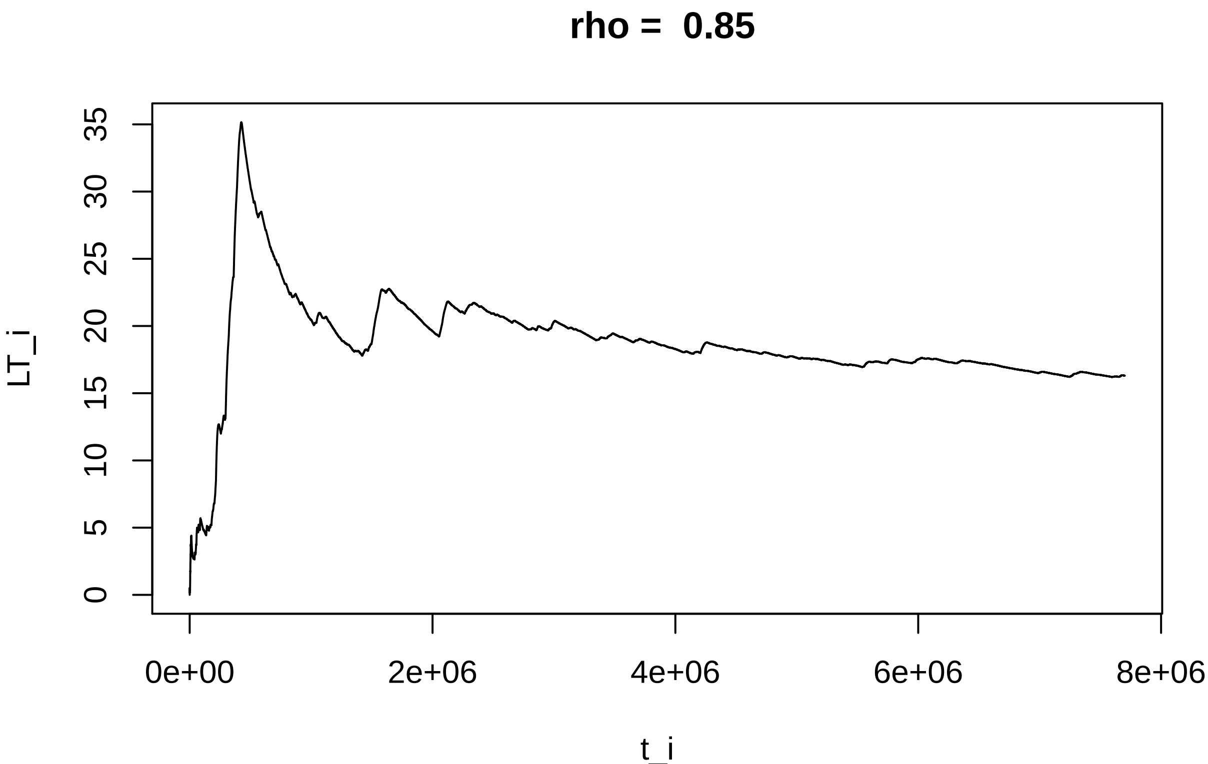
ρ	μ	W_q	L_q
0.4	2.4070657	69.1507968	0.8980623
0.7	2.9666815	423.5486306	5.5006316
0.85	3.1608375	1249.0362678	16.2212502
0.925	3.2453949	2958.3575269	38.4202276

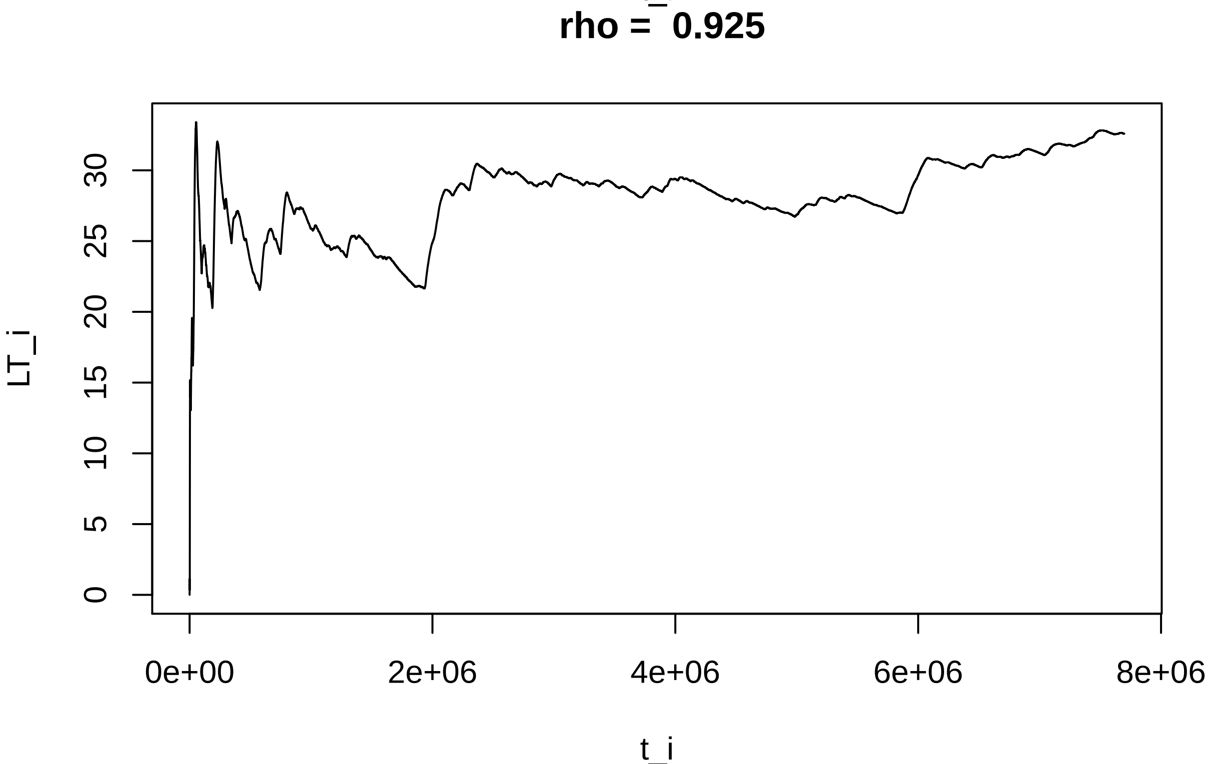
Simulation

First, for each ρ, we're going to calculate what is the amount of clients needed to get in the steady state of the waiting system.


rho = 0.4
LT_i
0.0
0.5
1.0
1.5
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

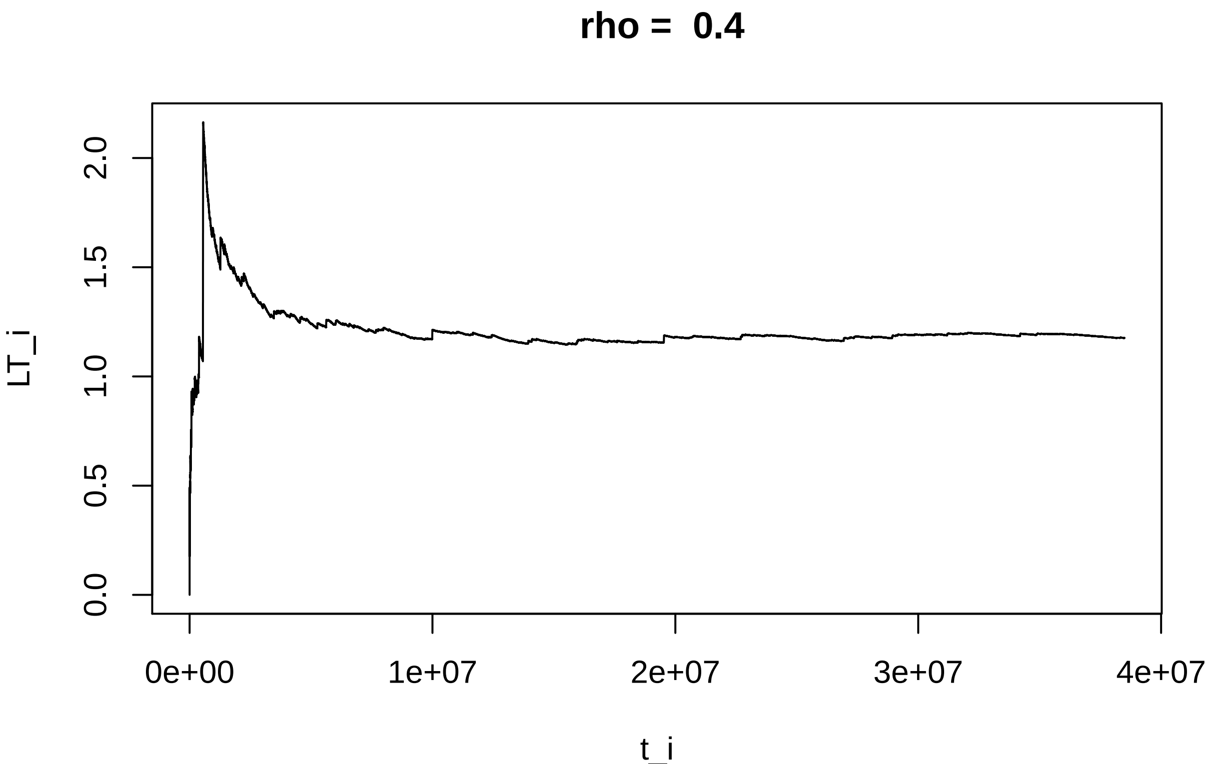

rho = 0.7
LT_i
0
2
4
6
8
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

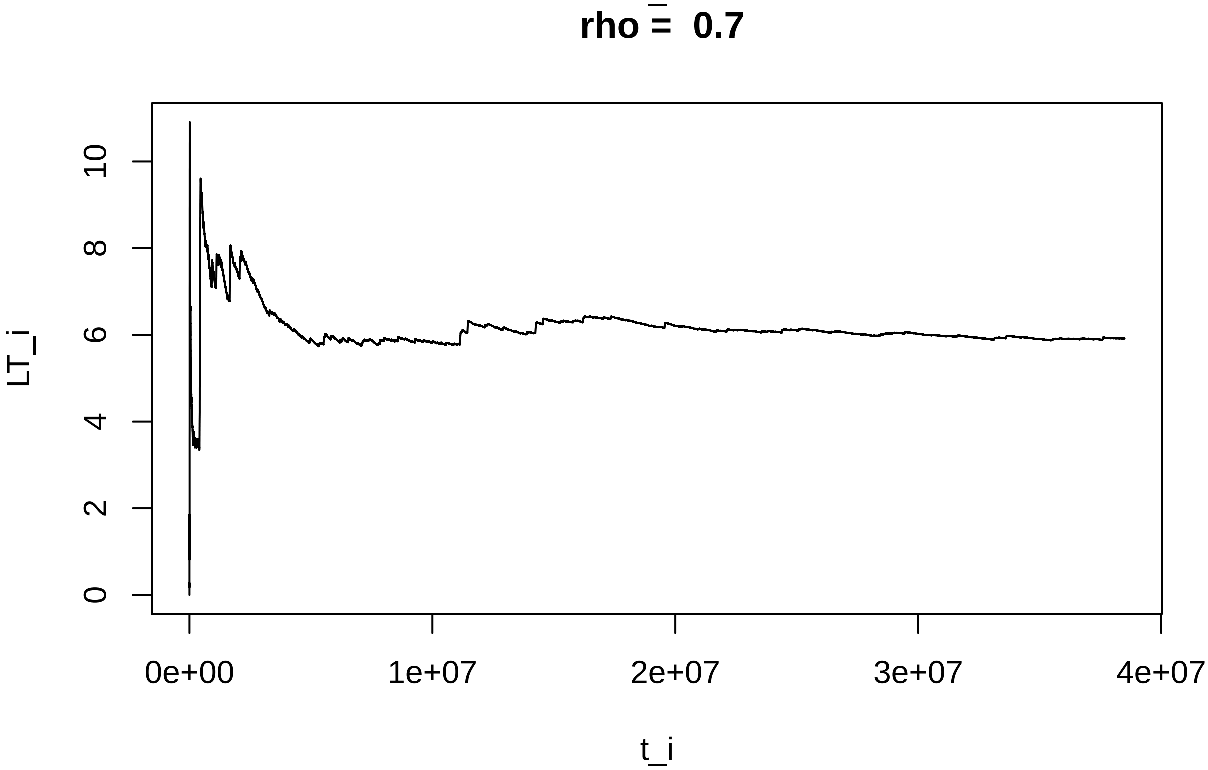


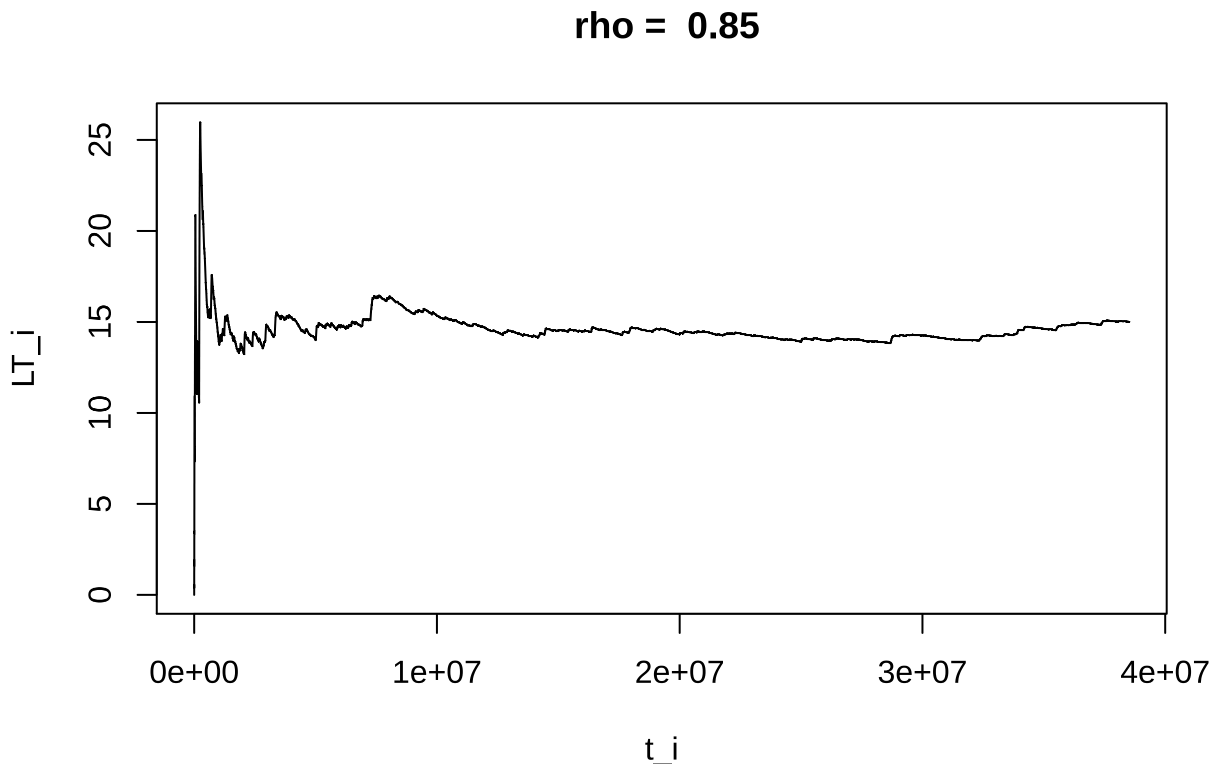


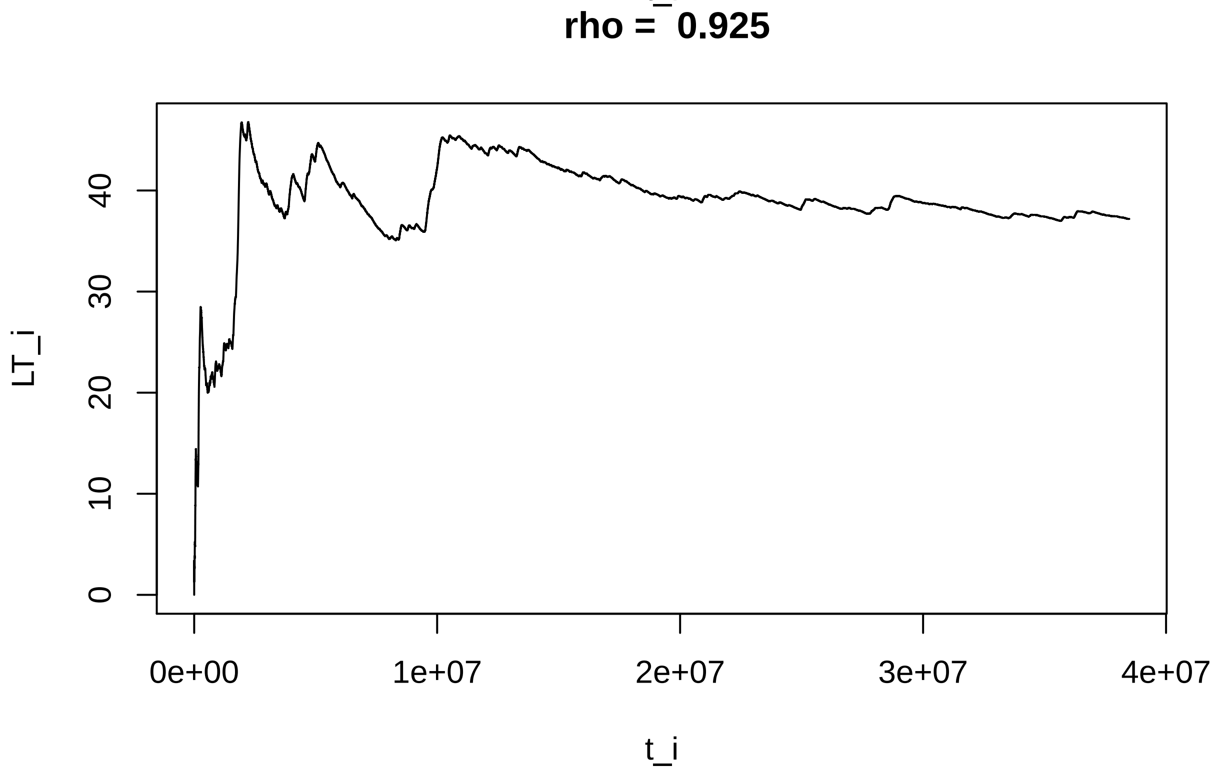
We observe that, apart from the simulation with loading factor 0.4, the other simulations have not attained the steady state.

If we repeat the simulations with a number of clients between 200000 and 500000, the steady state is attained with all loading factors. We have not tested more than 500000 clients.


rho = 0.4
LT_i
0.0
0.5
1.0
1.5
2.0
0e+00
1e+07
2e+07
3e+07
4e+07
t_i


rho = 0.7
LT_i
0
2
4
6
8
10
0e+00
1e+07
2e+07
3e+07
4e+07
t_i

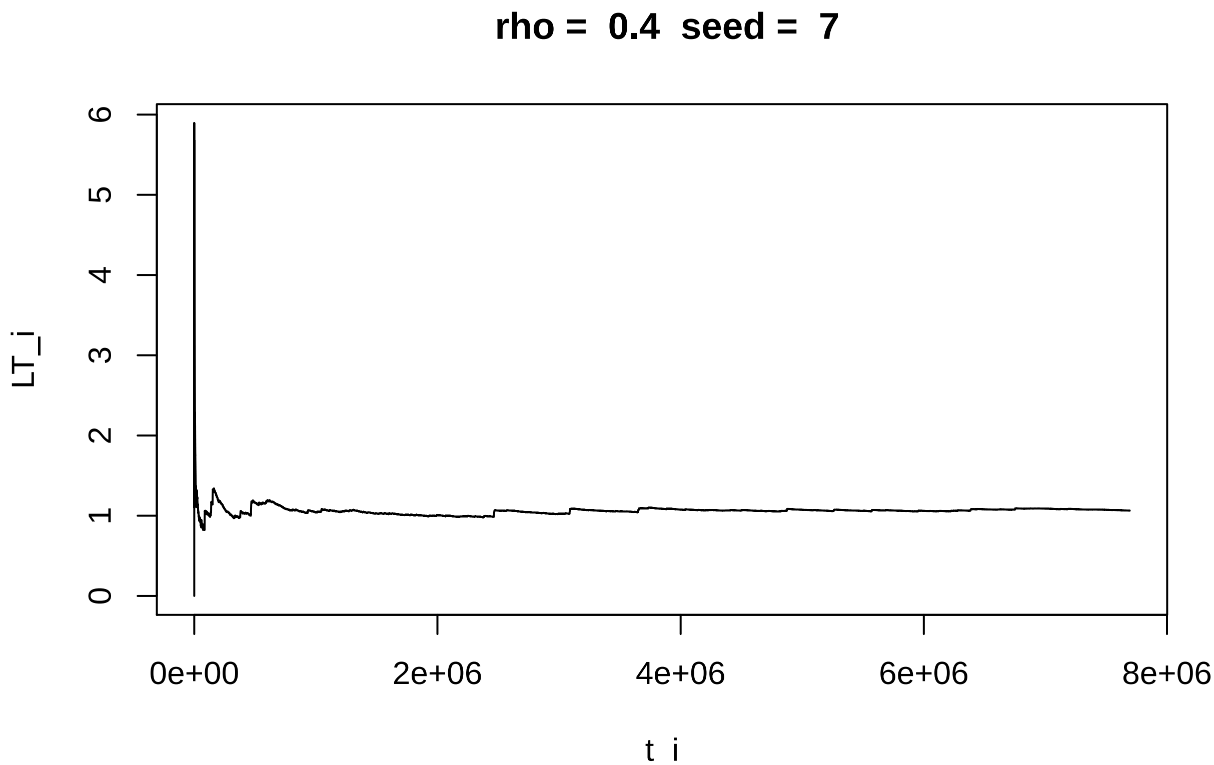


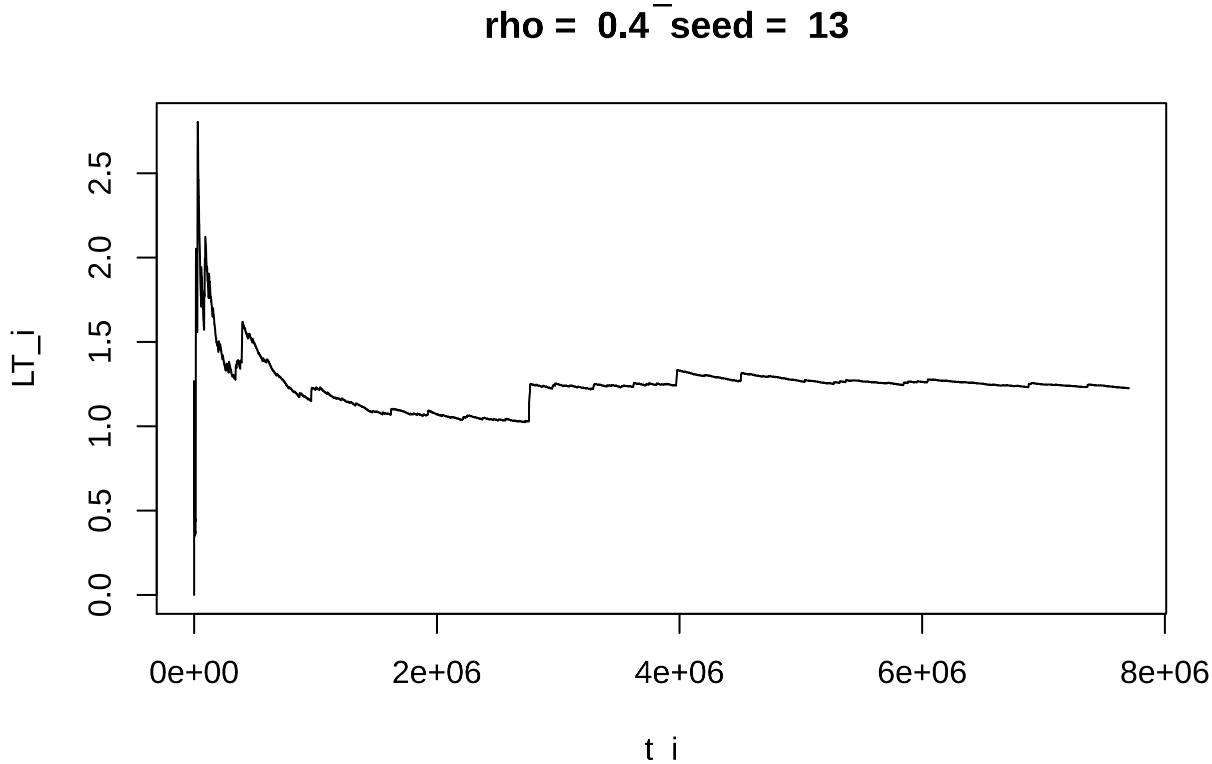


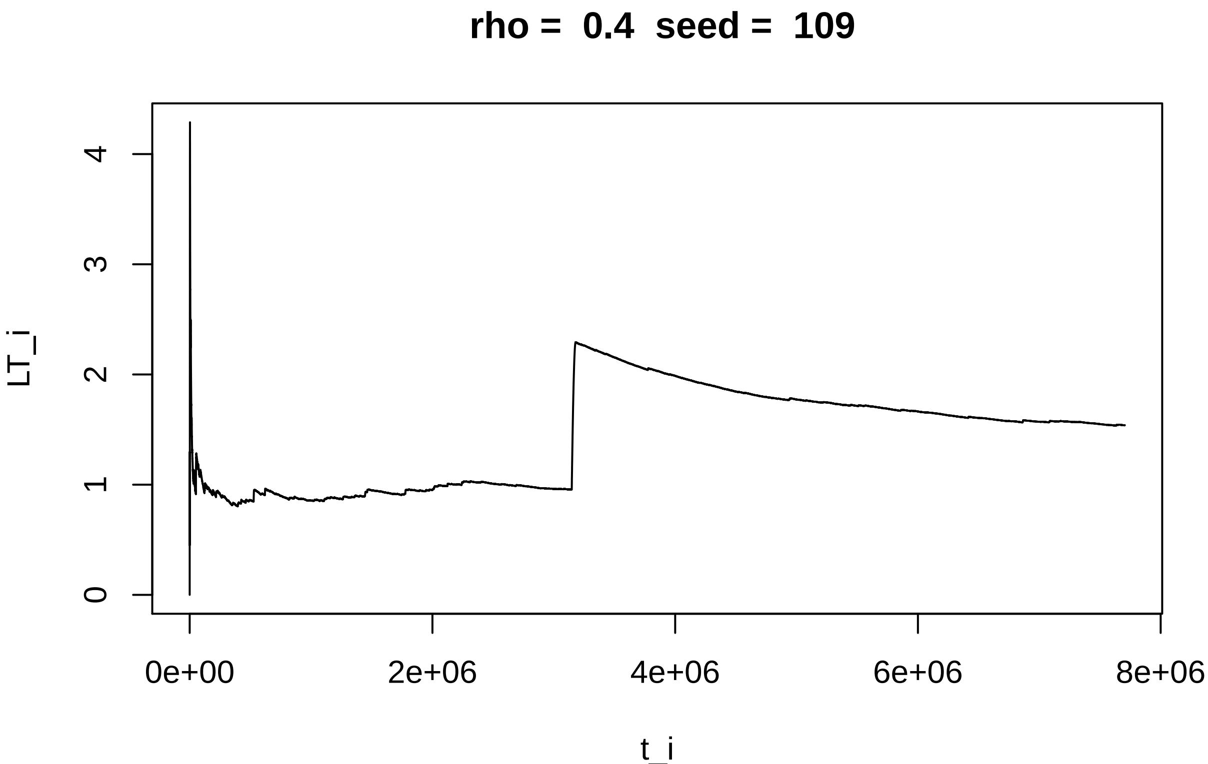
$\rho = 0.4$

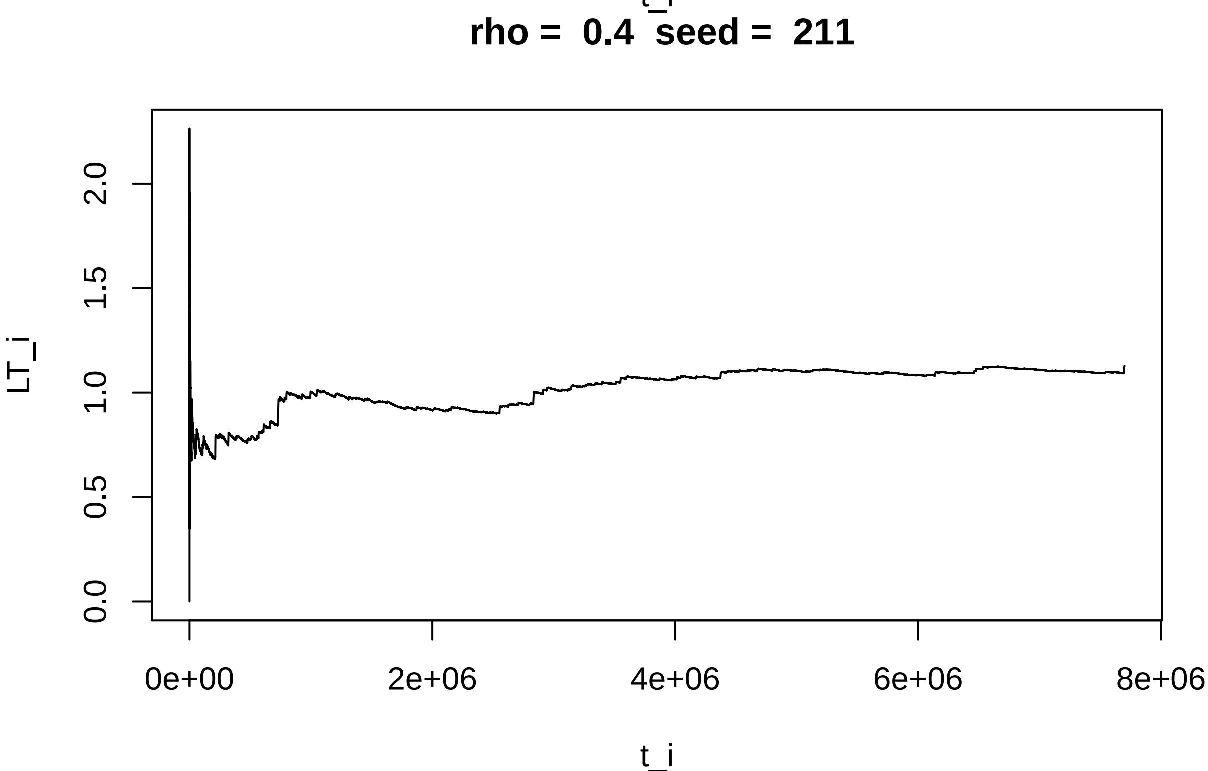
We generate 10 simulations with 100000 clients, each with a different seed. For each simulation, we check if the steady state is attained without any abrupt increase or decrease in the value of the average occupancy.

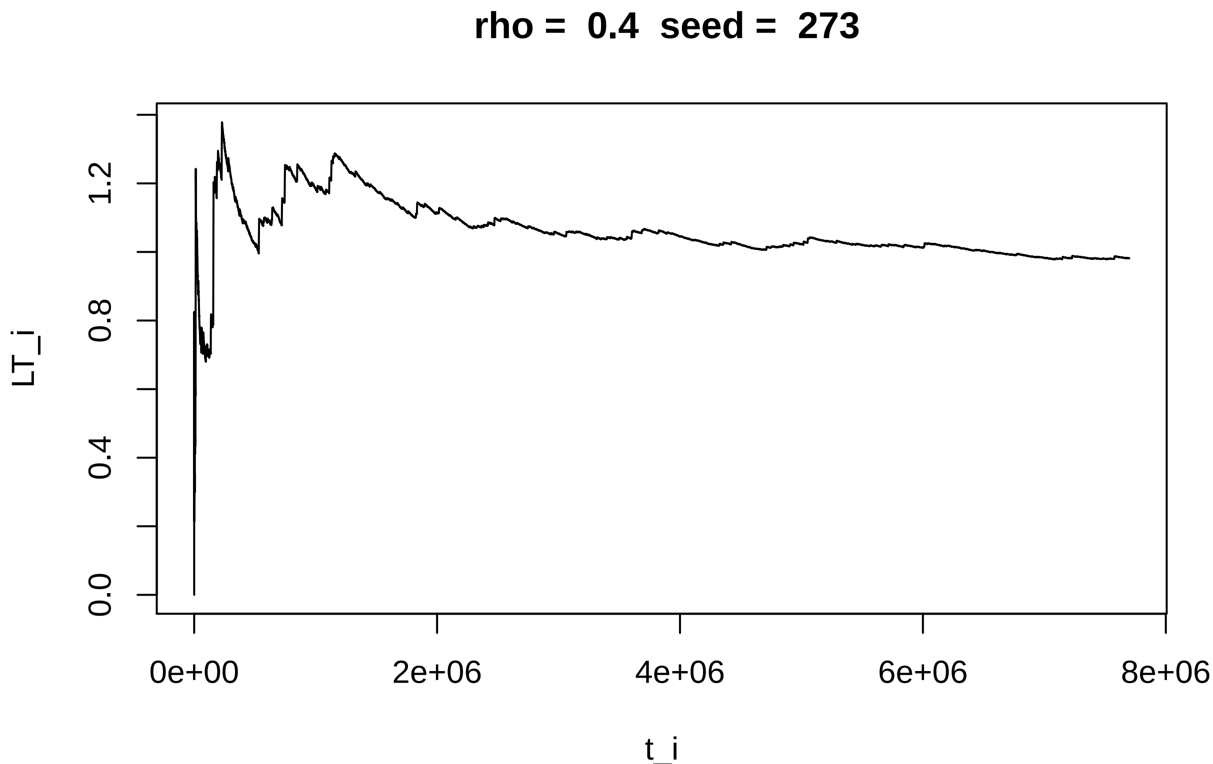
In case there's any abrupt increase or decrease, we change the seed. If for many seeds, this phenomena is still happening, we increase the number of clients.

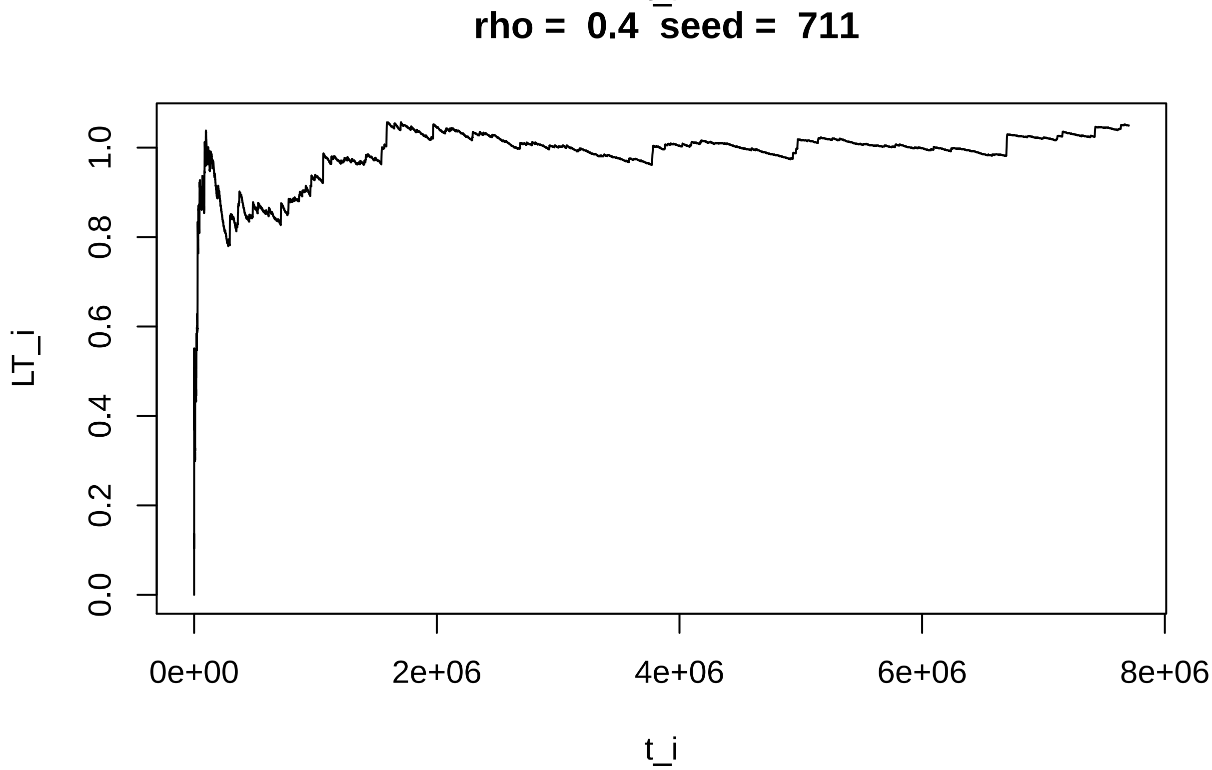


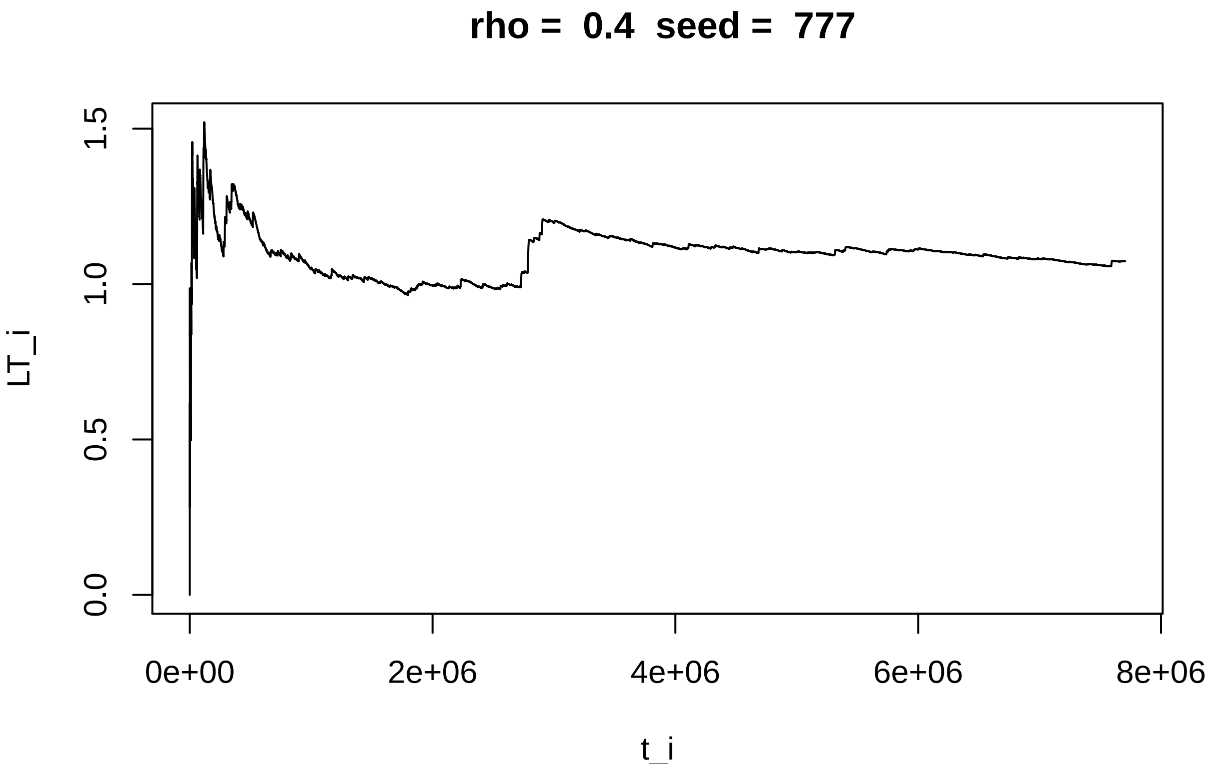


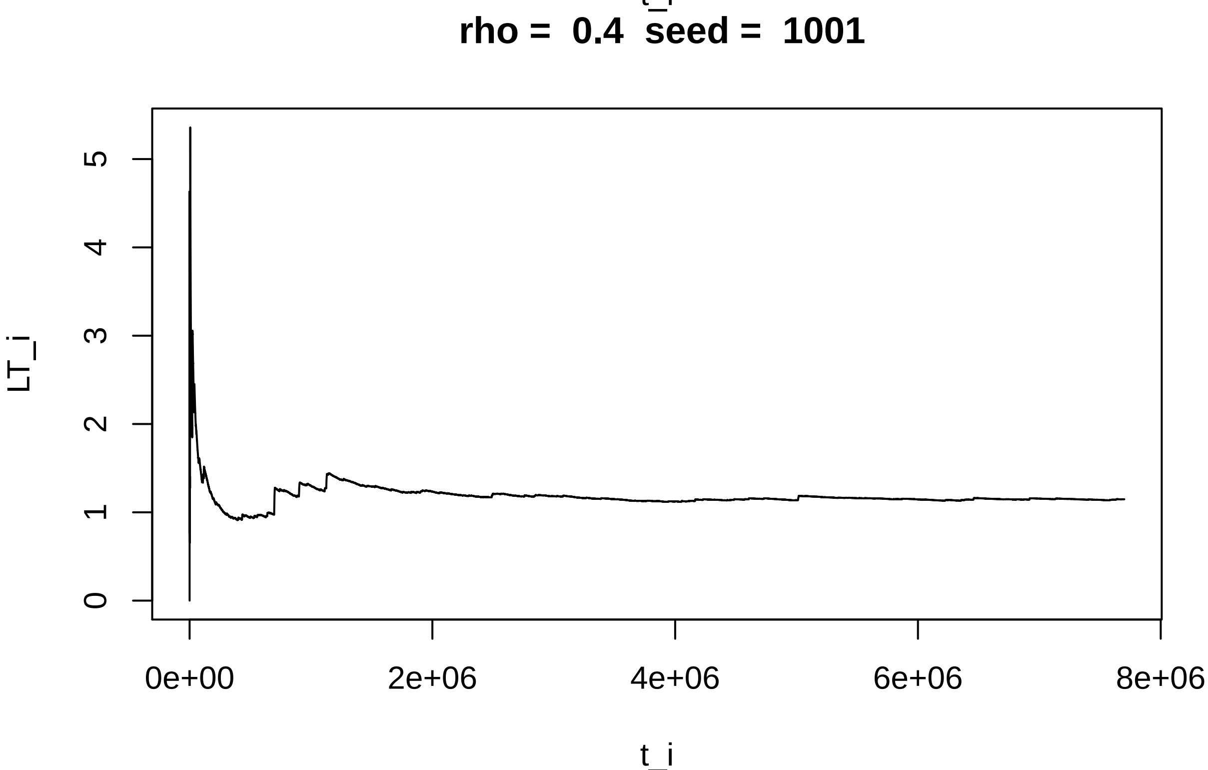

rho = 0.4 seed = 109
LT_i
0
1
2
3
4
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

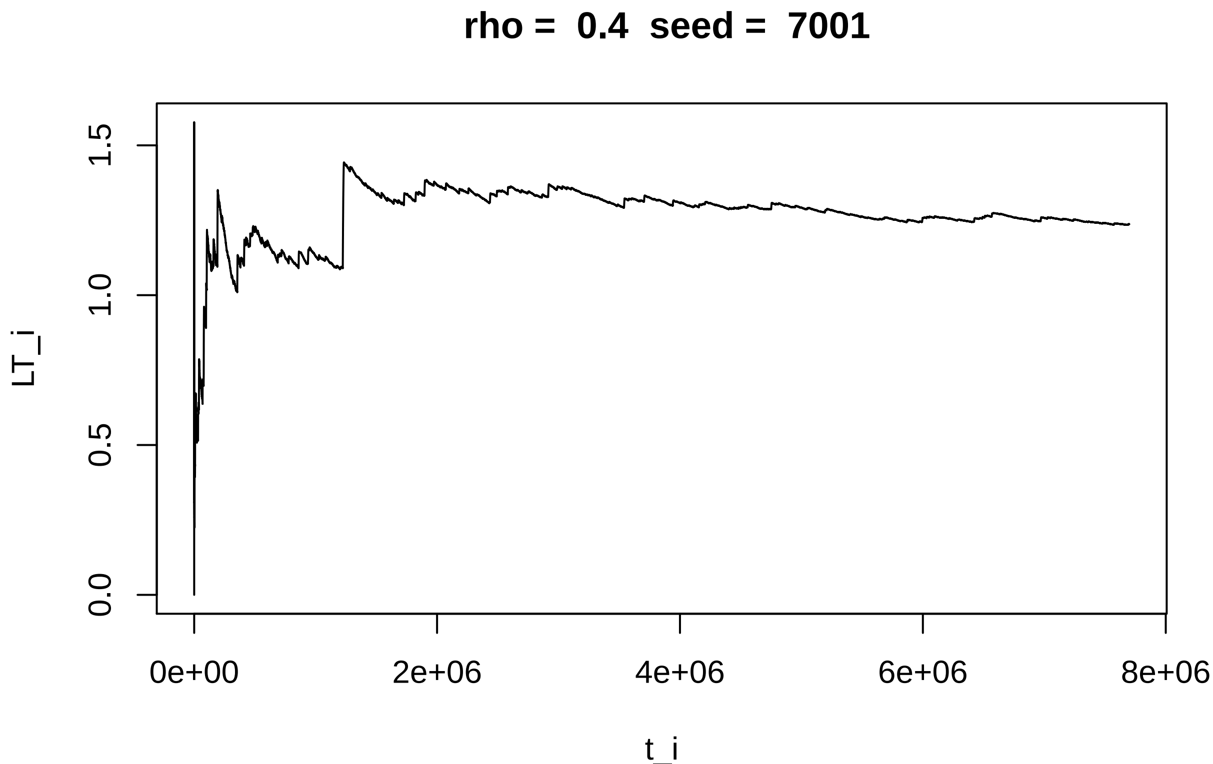

rho = 0.4 seed = 211
LT_i
0.0
0.5
1.0
1.5
2.0
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

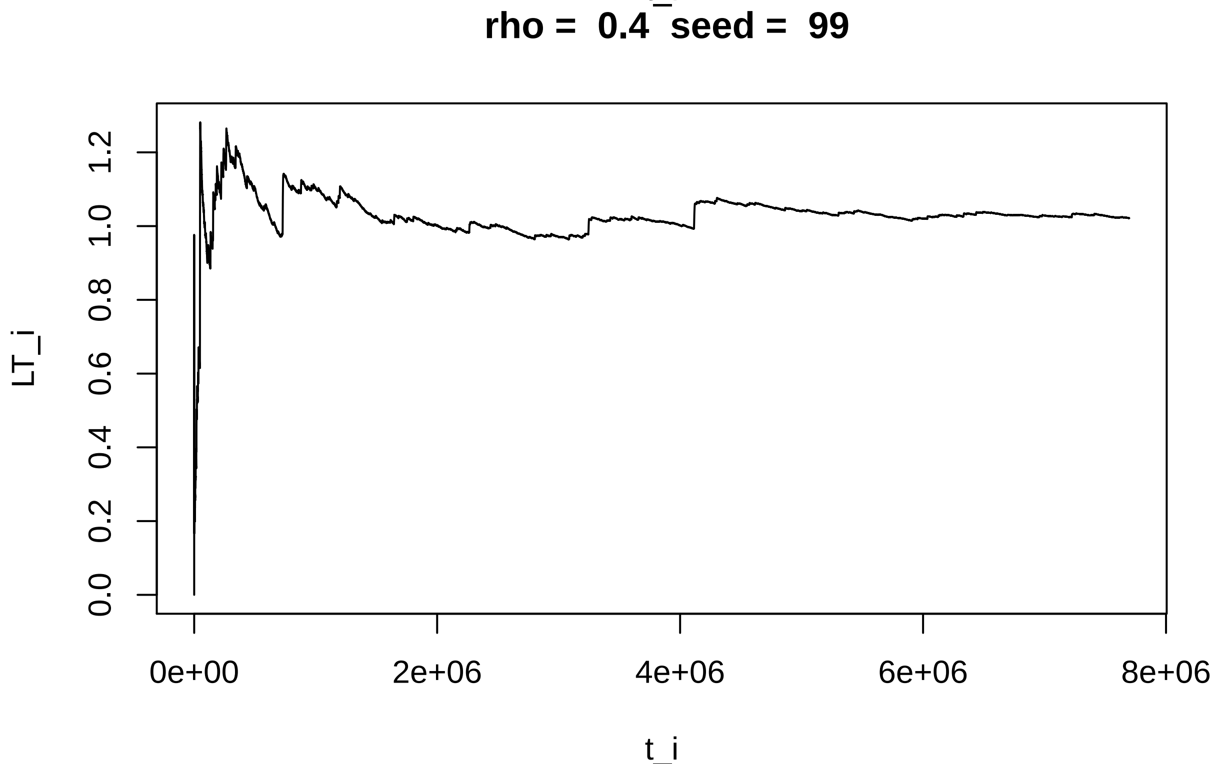

rho = 0.4 seed = 273
LT_i
t_i
0.0
0.4
0.8
1.2
0e+00
2e+06
4e+06
6e+06
8e+06


rho = 0.4 seed = 711
LT_i
t_i
0.0
0.2
0.4
0.6
0.8
1.0
0e+00
2e+06
4e+06
6e+06
8e+06

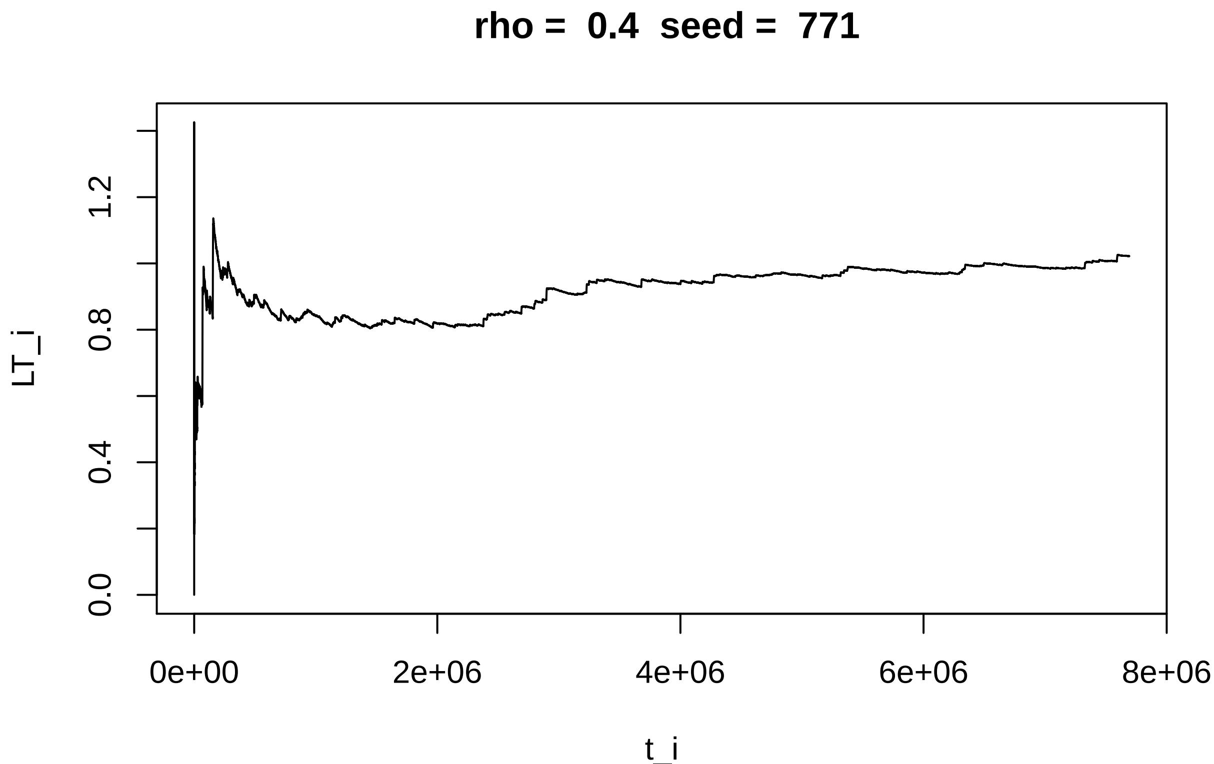

rho = 0.4 seed = 777
LT_i
t_i
0.0
0.5
1.0
1.5
0e+00
2e+06
4e+06
6e+06
8e+06

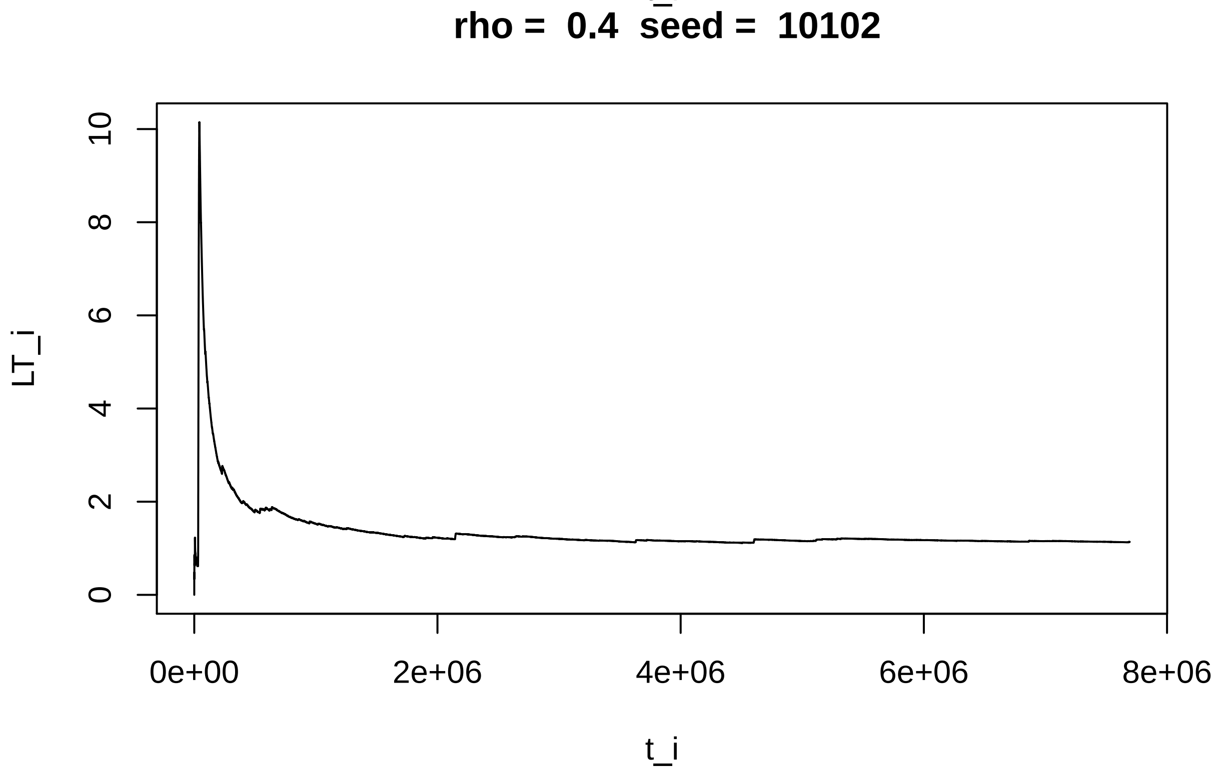

rho = 0.4 seed = 1001
LT_i
t_i
0
1
2
3
4
5
0e+00
2e+06
4e+06
6e+06
8e+06

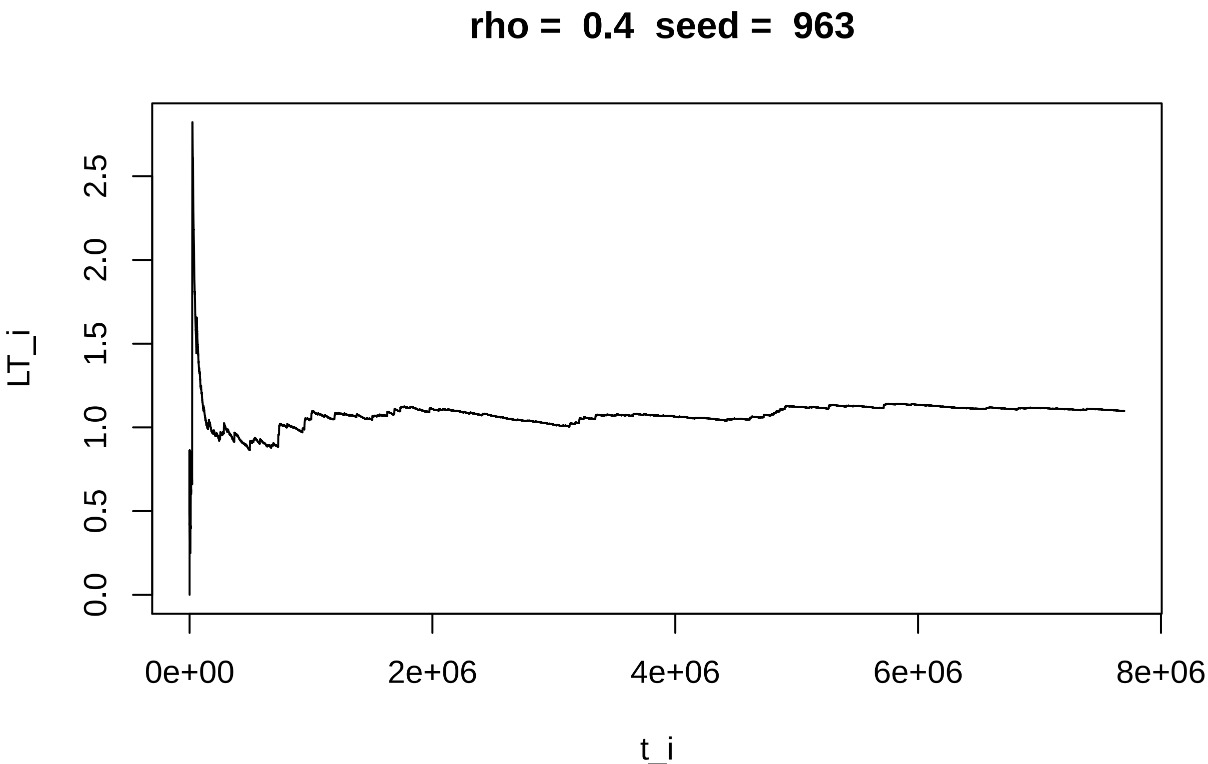


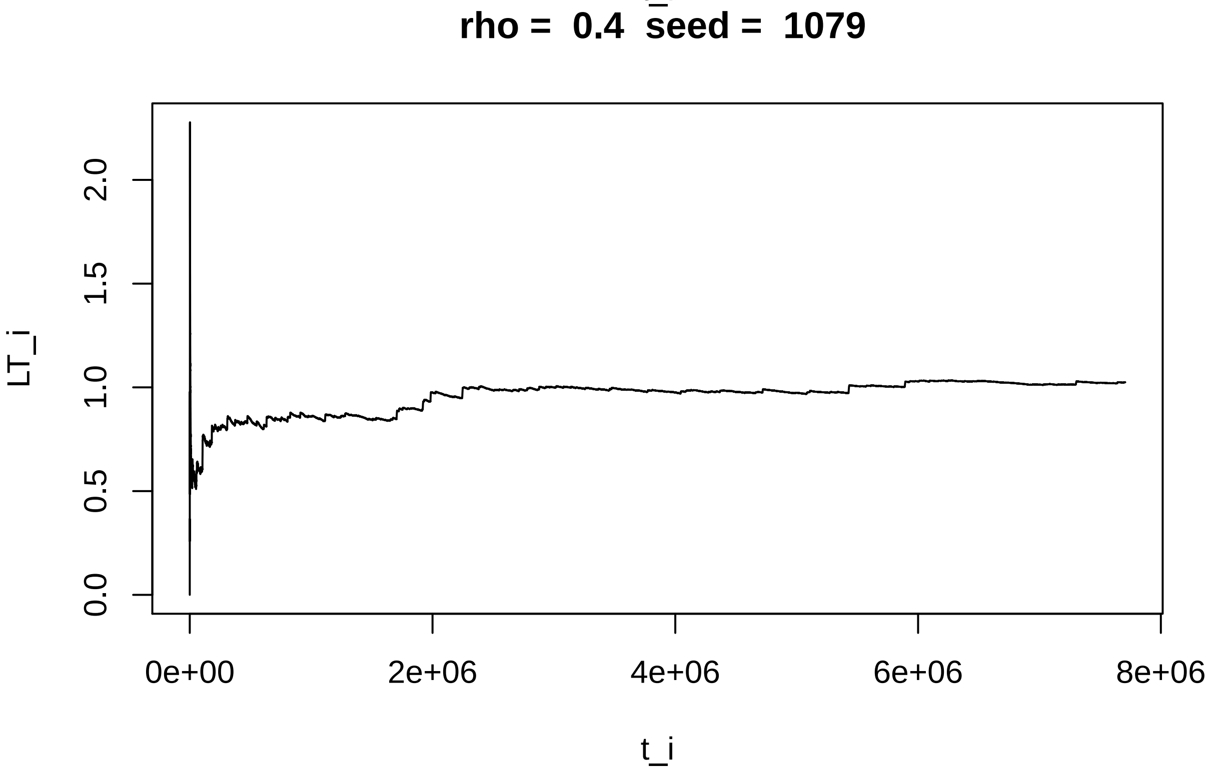


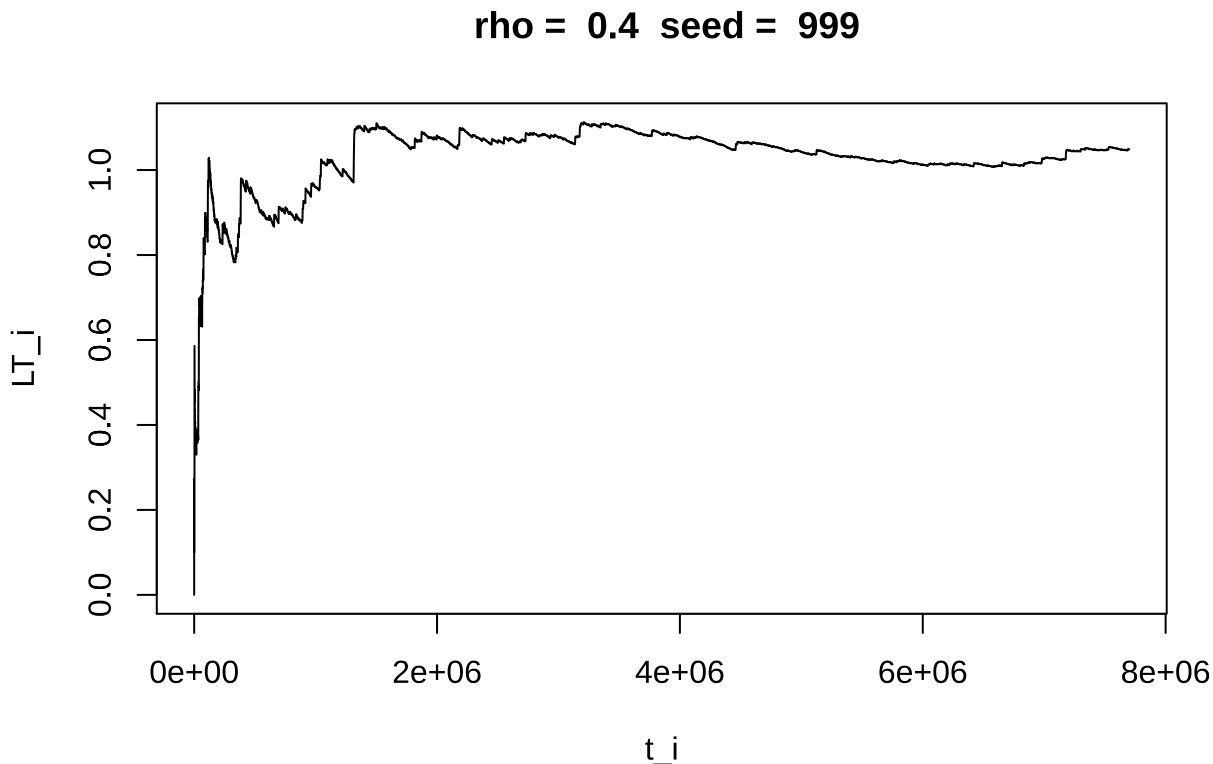
We observe that for the seeds 10101, 1078, 960 and 51, there's an abrupt change in the average occupancy. We change those seeds and redo the simulation.

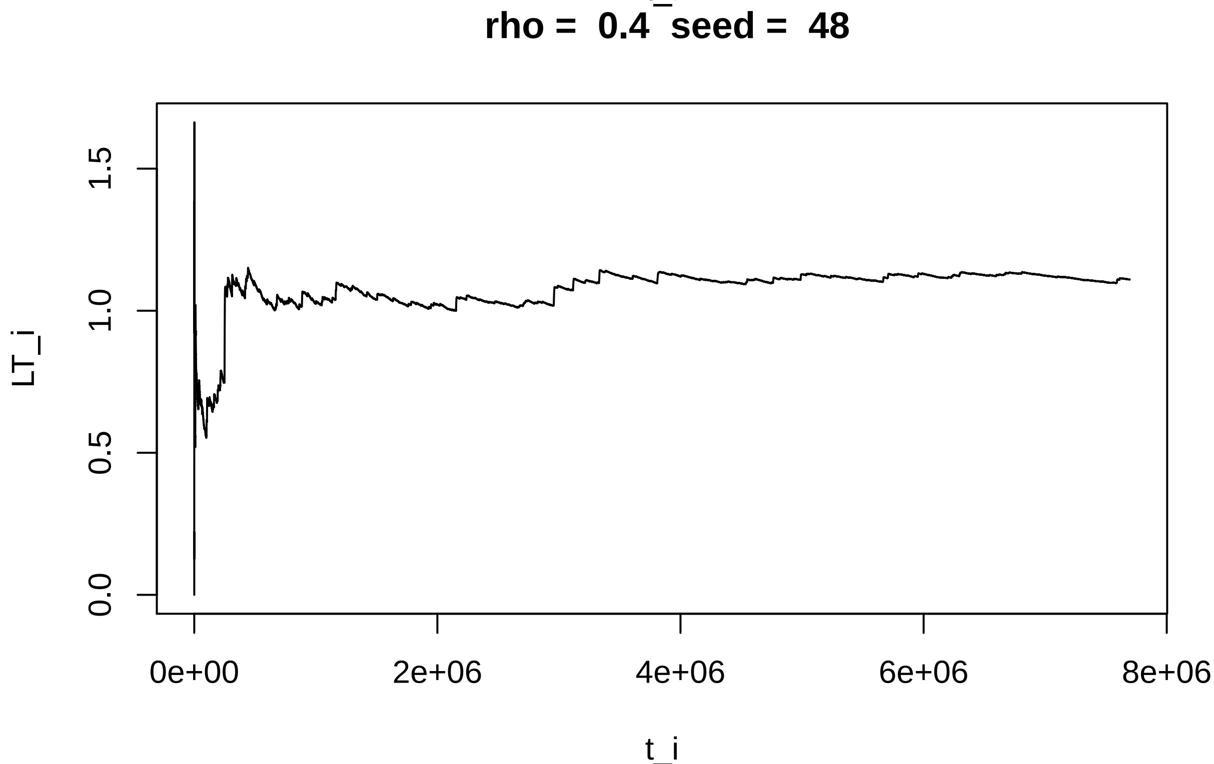

rho = 0.4 seed = 771
LT_i
0.0
0.4
0.8
1.2
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

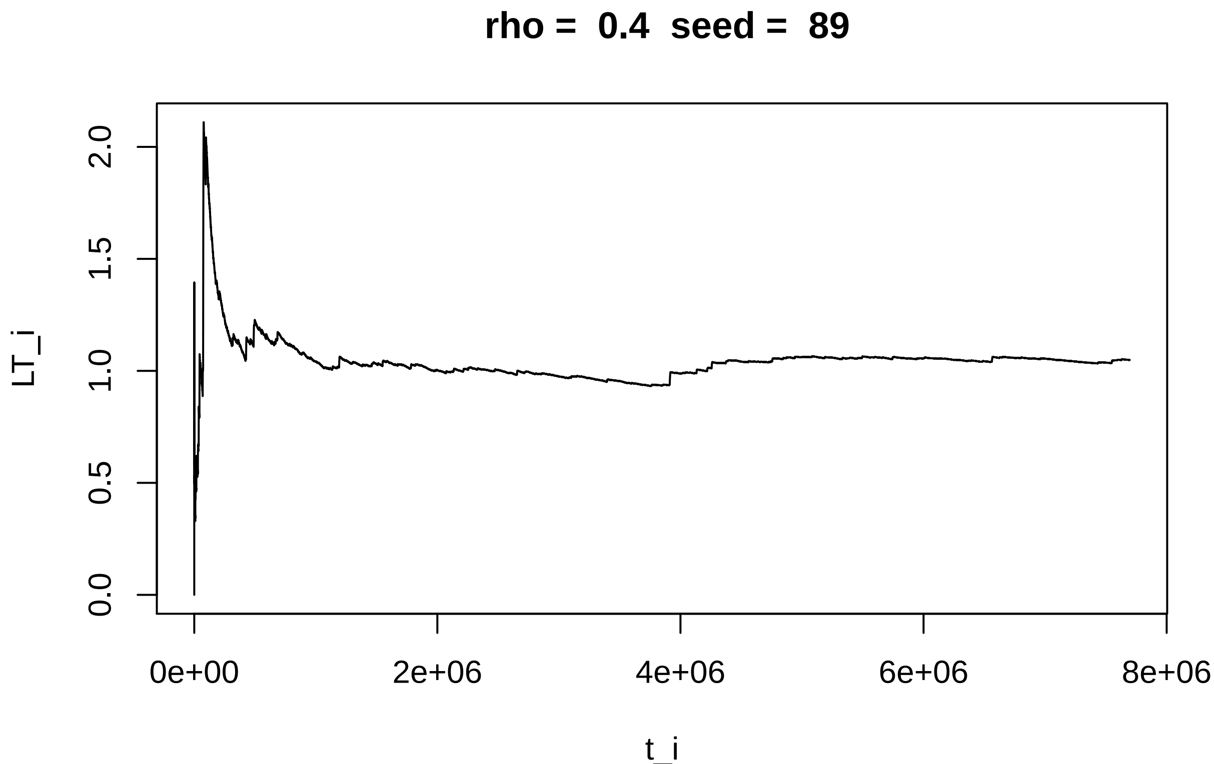

rho = 0.4 seed = 10102
LT_i
0
2
4
6
8
10
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

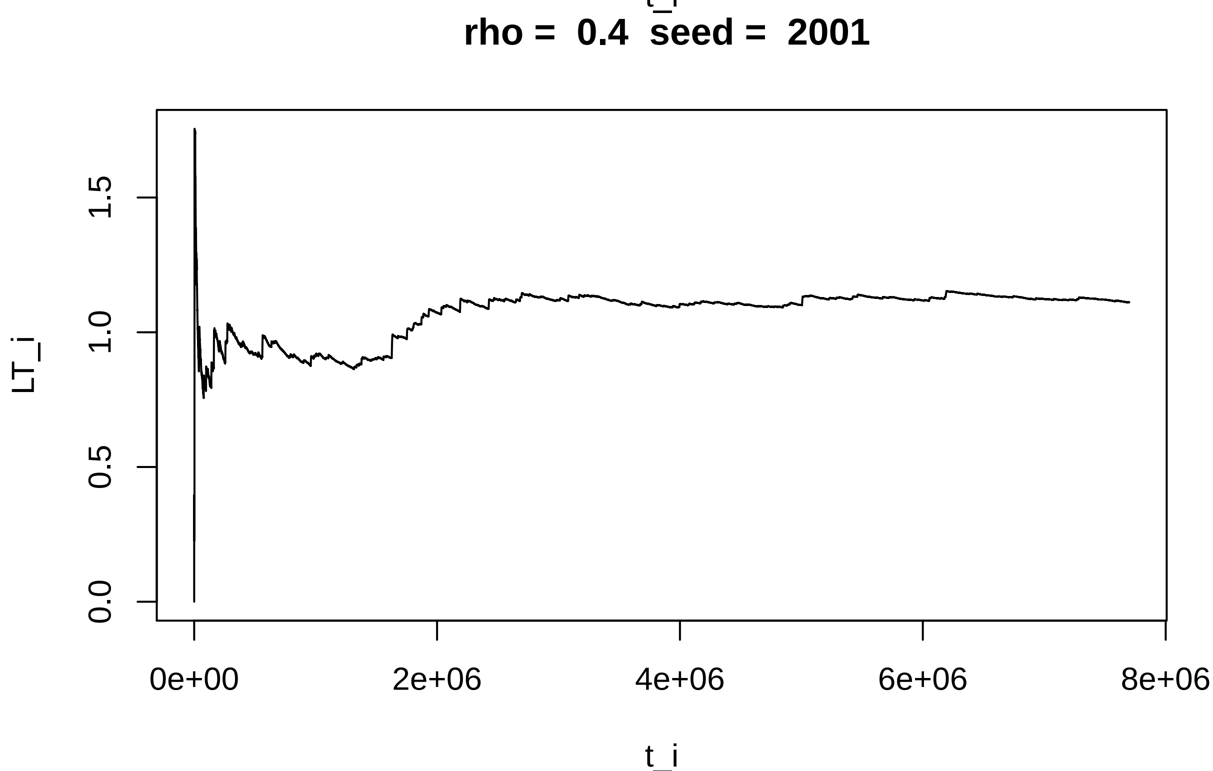

rho = 0.4 seed = 963
LT_i
t_i
0.0
0.5
1.0
1.5
2.0
2.5
0e+00
2e+06
4e+06
6e+06
8e+06

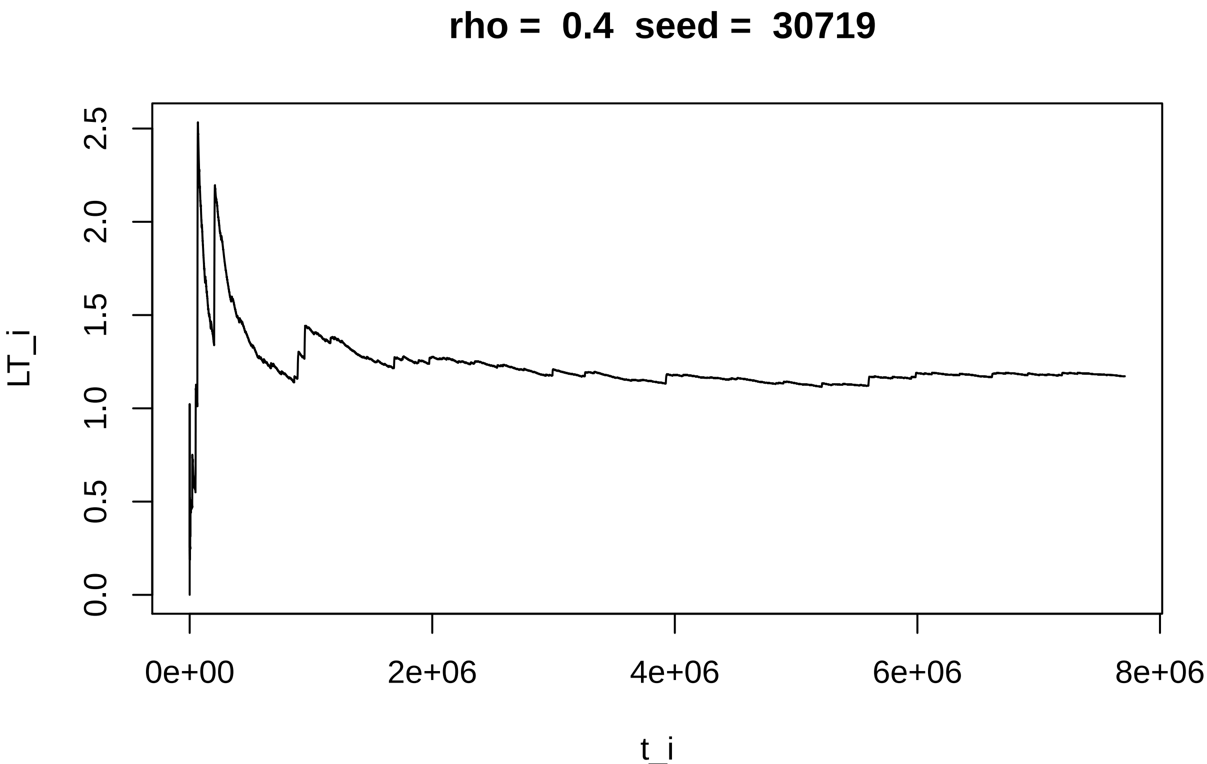

rho = 0.4 seed = 1079
LT_i
t_i
0.0
0.5
1.0
1.5
2.0
0e+00
2e+06
4e+06
6e+06
8e+06

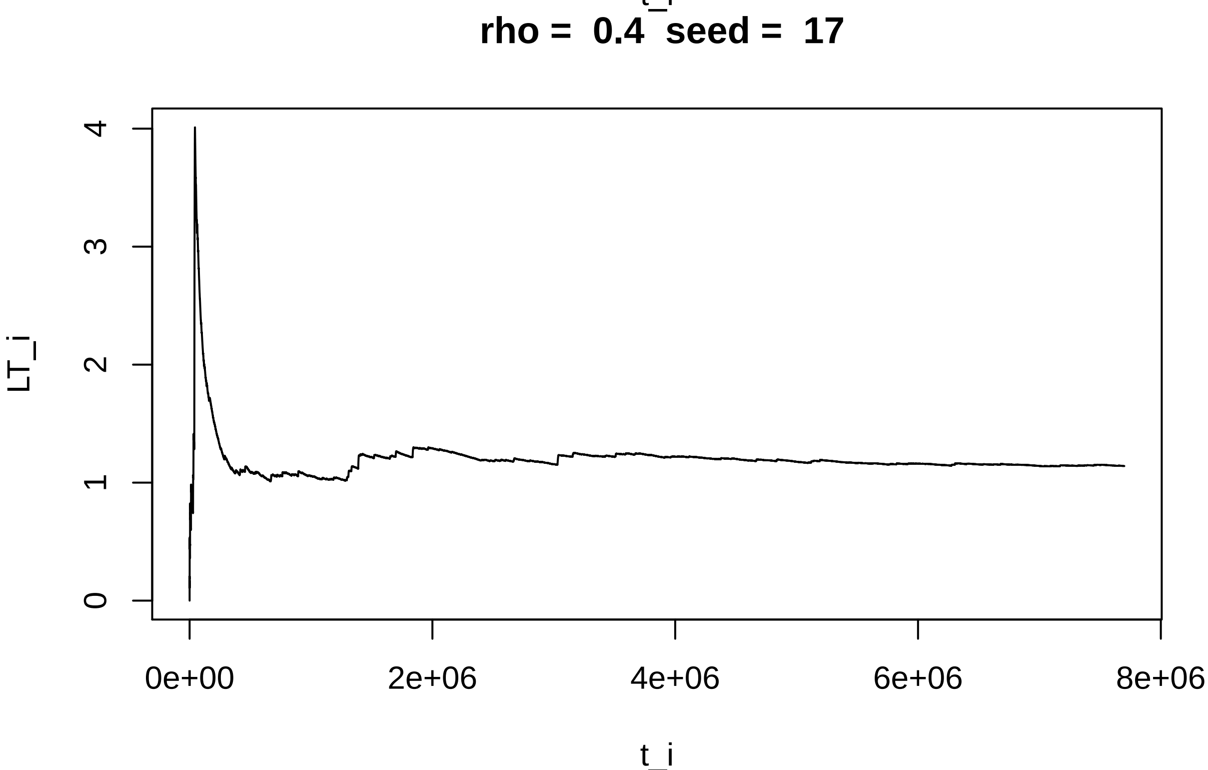

rho = 0.4 seed = 999
LT_i
0.0
0.2
0.4
0.6
0.8
1.0
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.4 seed = 48
LT_i
0.0
0.5
1.0
1.5
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.4 seed = 89
LT_i
0.0
0.5
1.0
1.5
2.0
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.4 seed = 2001
LT_i
0.0
0.5
1.0
1.5
0e+00
2e+06
4e+06
6e+06
8e+06
t_i





We changed a total of 4 out of 10 seeds.

We now compute the confidence interval for L_q and W_q. We will use a t-Student distribution with critical value 1 - 0.95 and 9 degrees of freedom to compute a 95% confidence interval.

$$L_{q_1}, L_{q_2}, ..., L_{q_{10}} \overline{L}_q = \frac{1}{n}\sum_{i=1}^{n} L_{q_i} S_{L_q}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(L_{q_i} - \overline{L}_q)^2 C.I.(L_q) = \overline{L}_q \pm t_{1-\alpha,n-1} \cdot \sqrt{\frac{S_{L_q}^2}{n}}$$

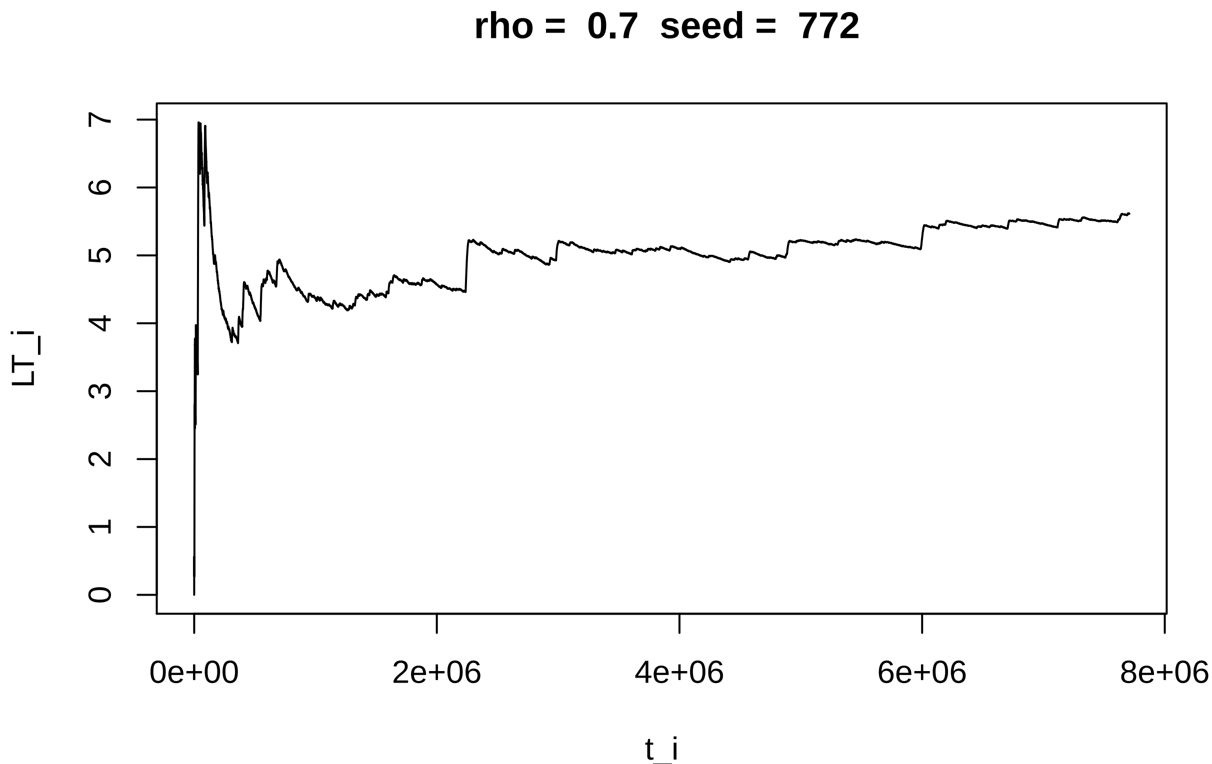
$$W_{q_1}, W_{q_2}, ..., W_{q_{10}} \overline{W}_q = \frac{1}{n}\sum_{i=1}^{n} W_{q_i} S_{W_q}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(W_{q_i} - \overline{W}_q)^2 C.I.(W_q) = \overline{W}_q \pm t_{1-\alpha,n-1} \cdot \sqrt{\frac{S_{W_q}^2}{n}}$$

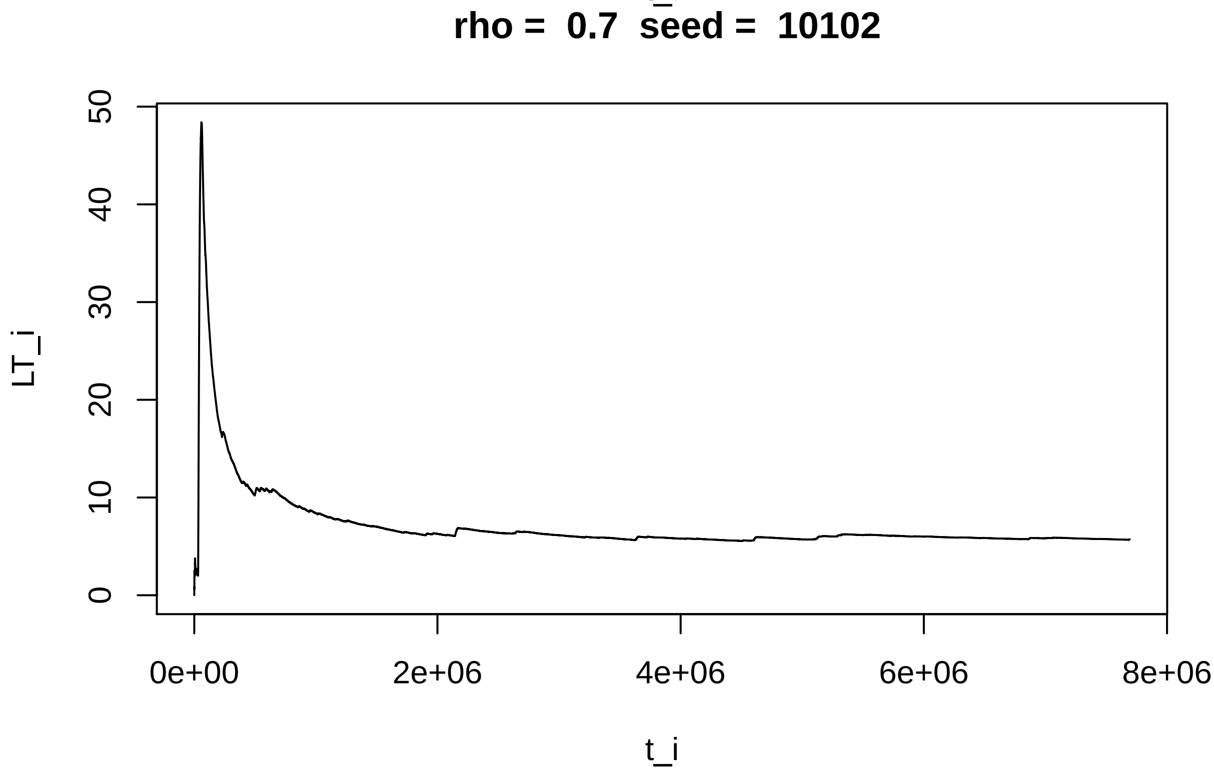
The computations produce the followings confidence intervals for the average queue length and waiting time:

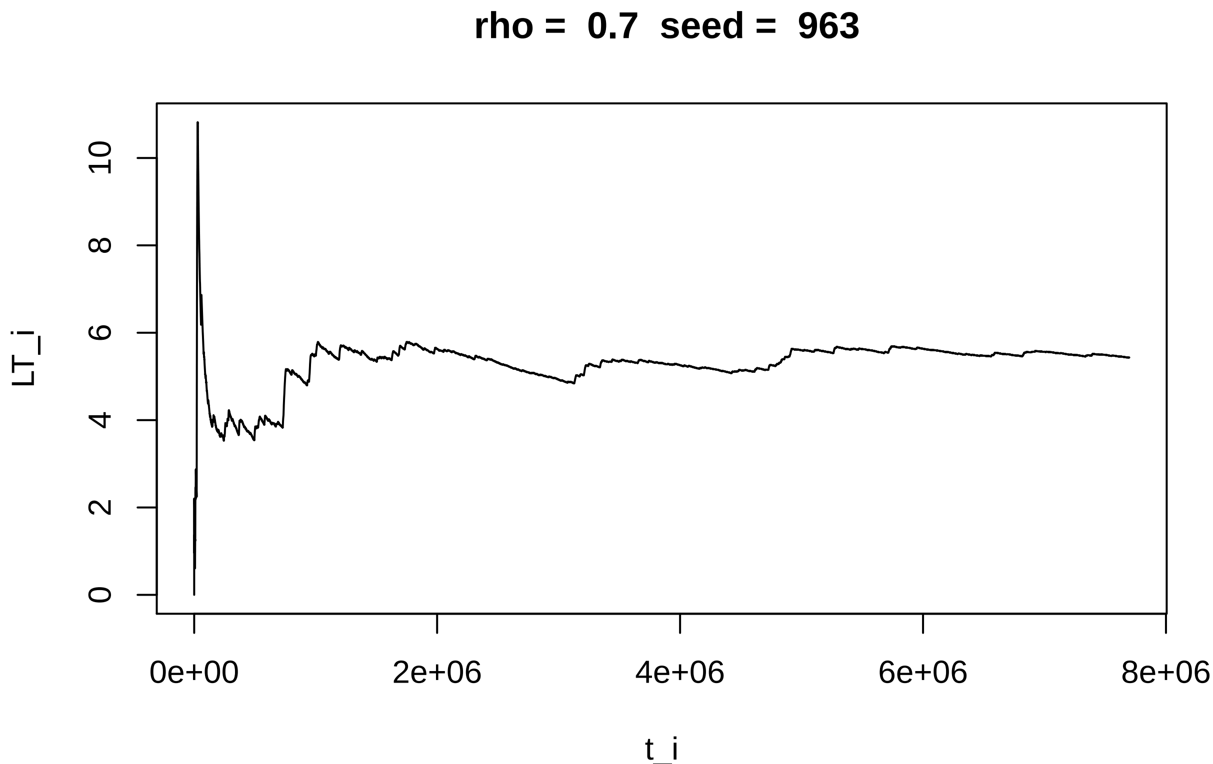
```
##    NA        NA        NA       NA      NA
## 1 0.4 0.6629266 0.7216626 51.03537 55.5688
```

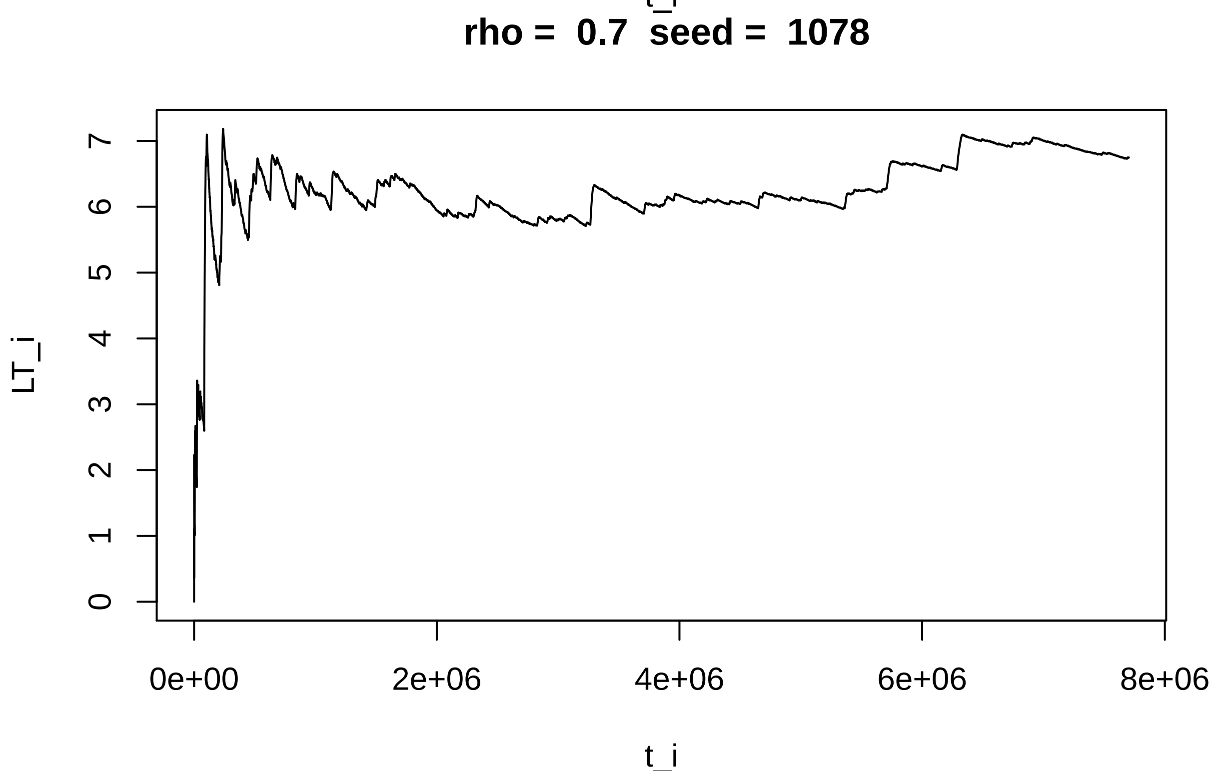
$\rho = \mathbf{0.7}$

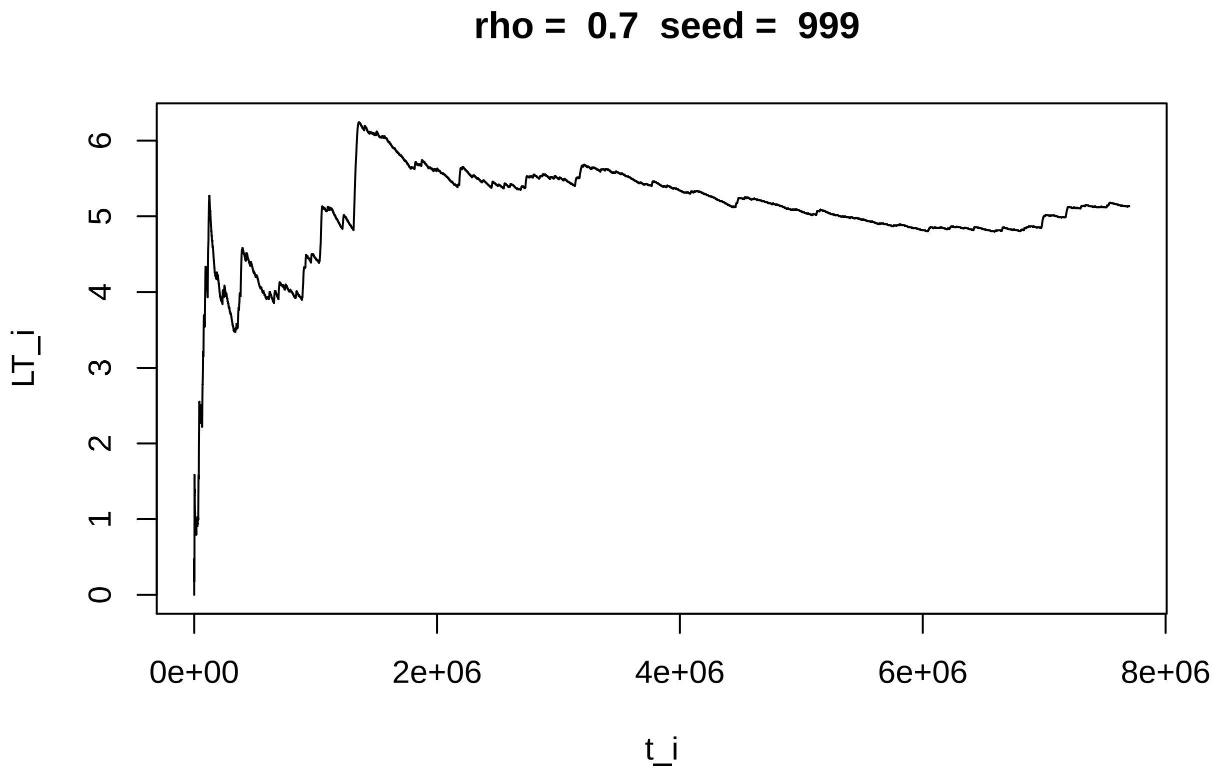
We generate 10 simulations with 100000 clients, each with a different seed. We check if the steady state is attained. We also check for irregularities in the simulation results, in which case we change the random number generator seed and repeat the simulation.

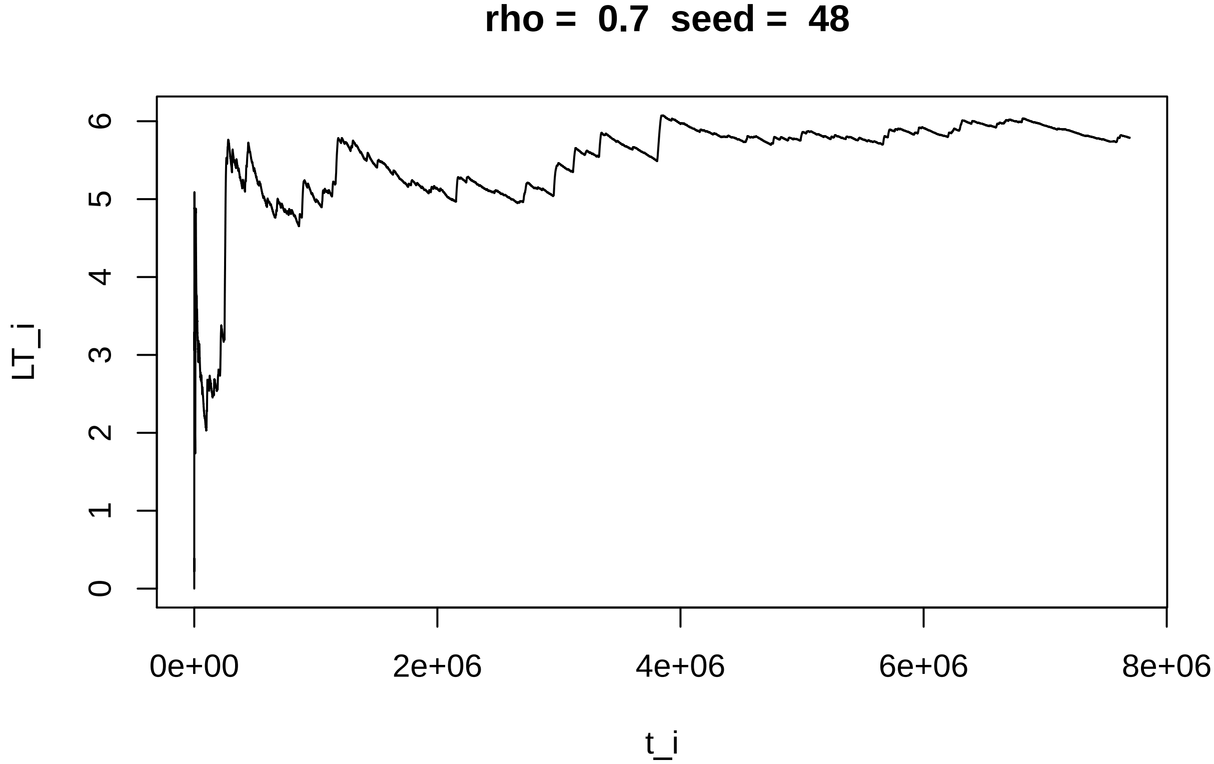

rho = 0.7 seed = 772
LT_i
t_i
0
1
2
3
4
5
6
7
0e+00
2e+06
4e+06
6e+06
8e+06

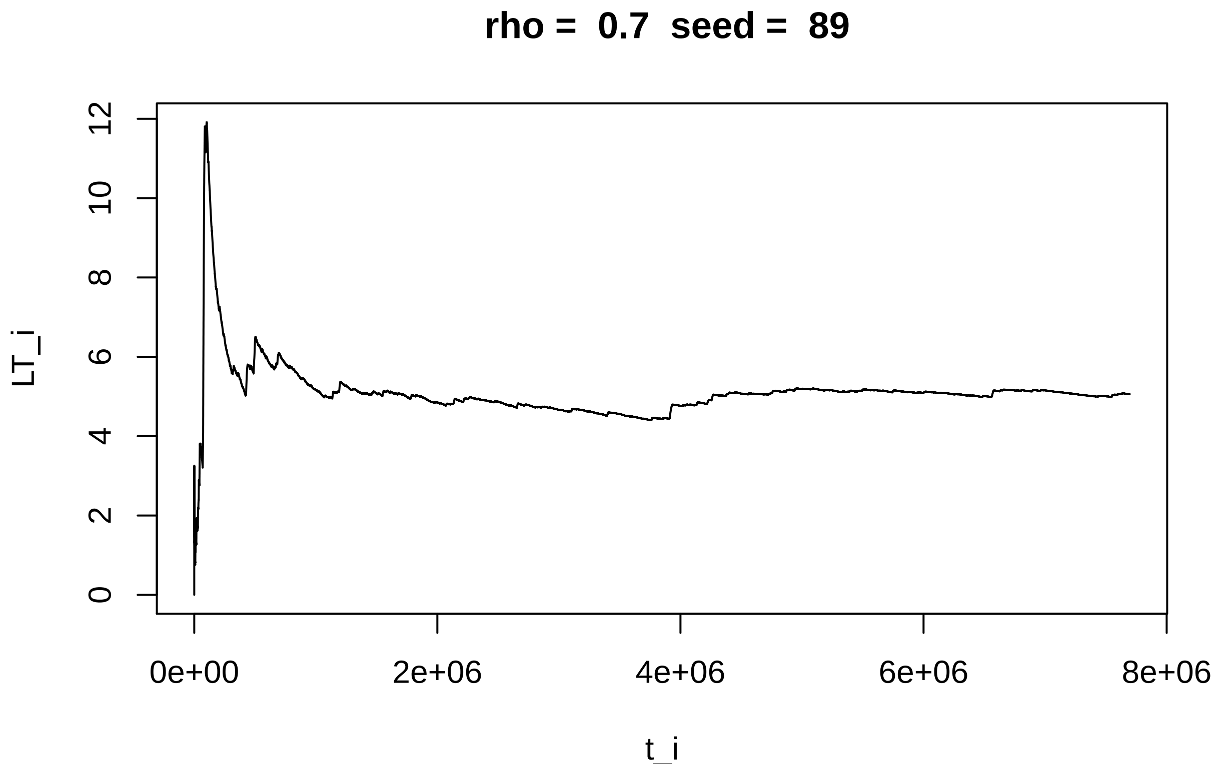

rho = 0.7 seed = 10102
LT_i
t_i
0
10
20
30
40
50
0e+00
2e+06
4e+06
6e+06
8e+06

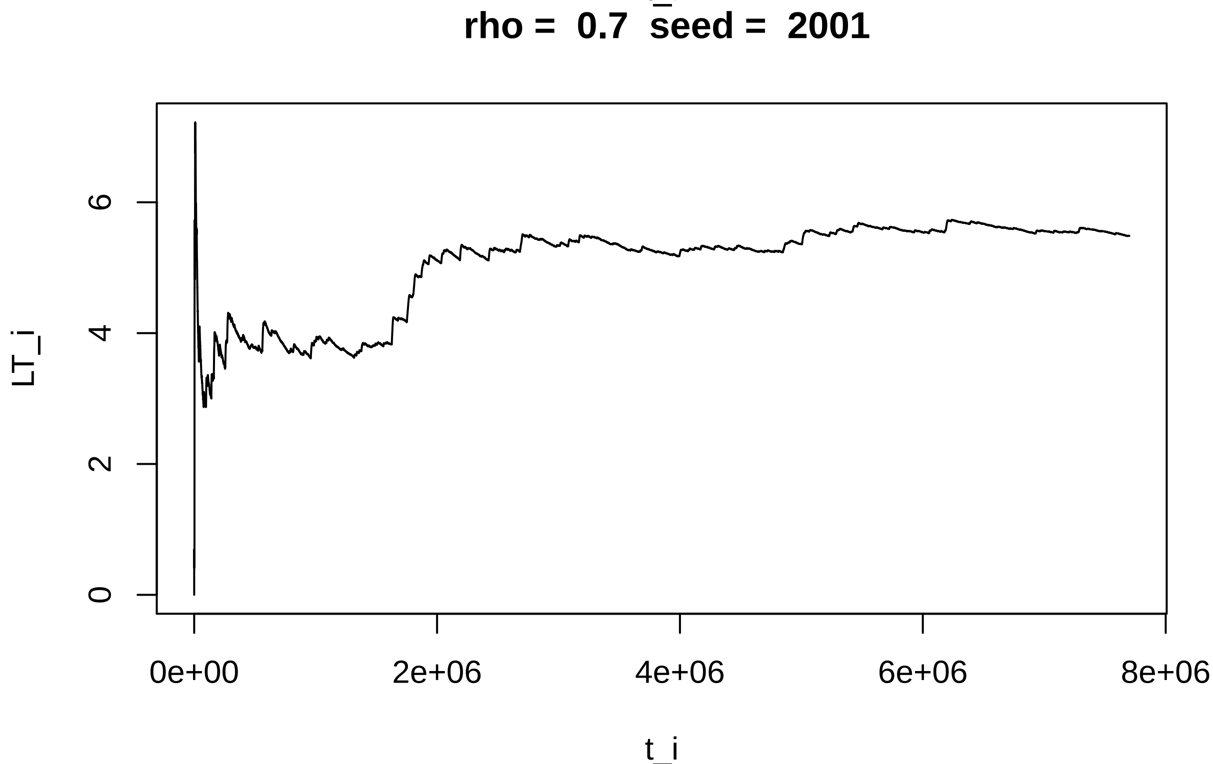

rho = 0.7 seed = 963
LT_i
0
2
4
6
8
10
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

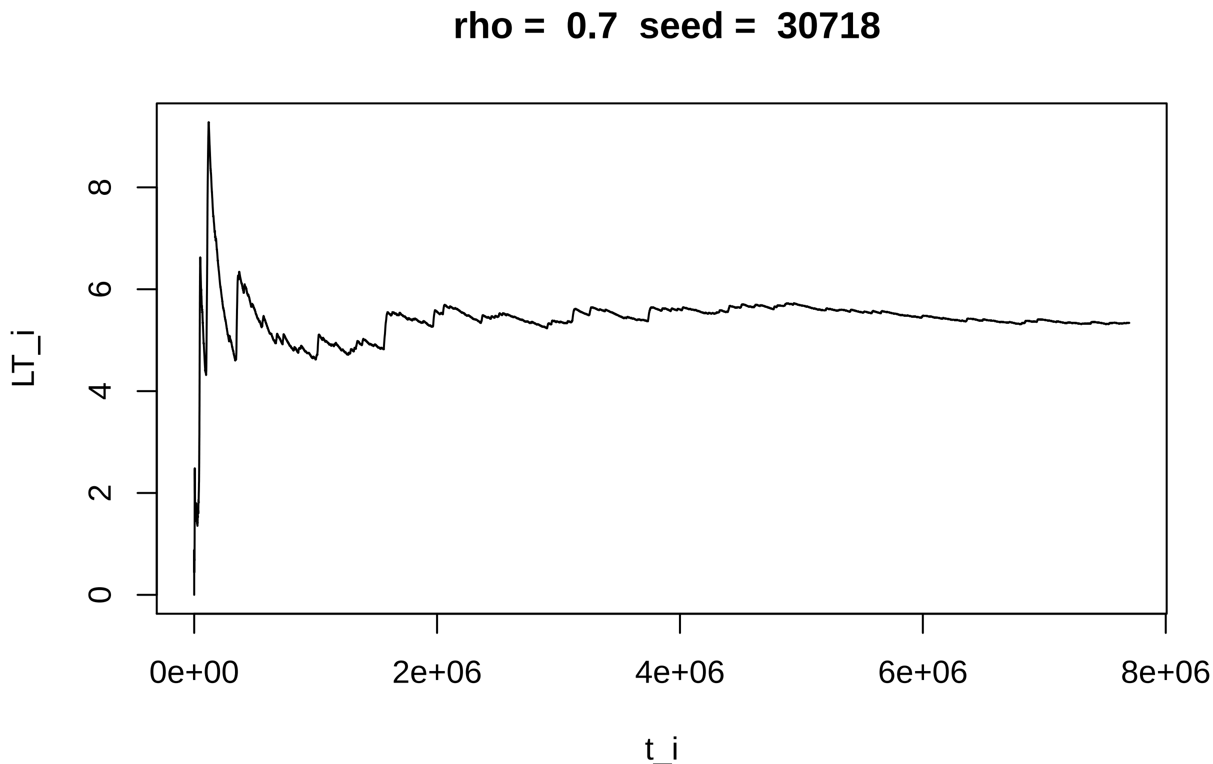

rho = 0.7 seed = 1078
LT_i
0
1
2
3
4
5
6
7
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

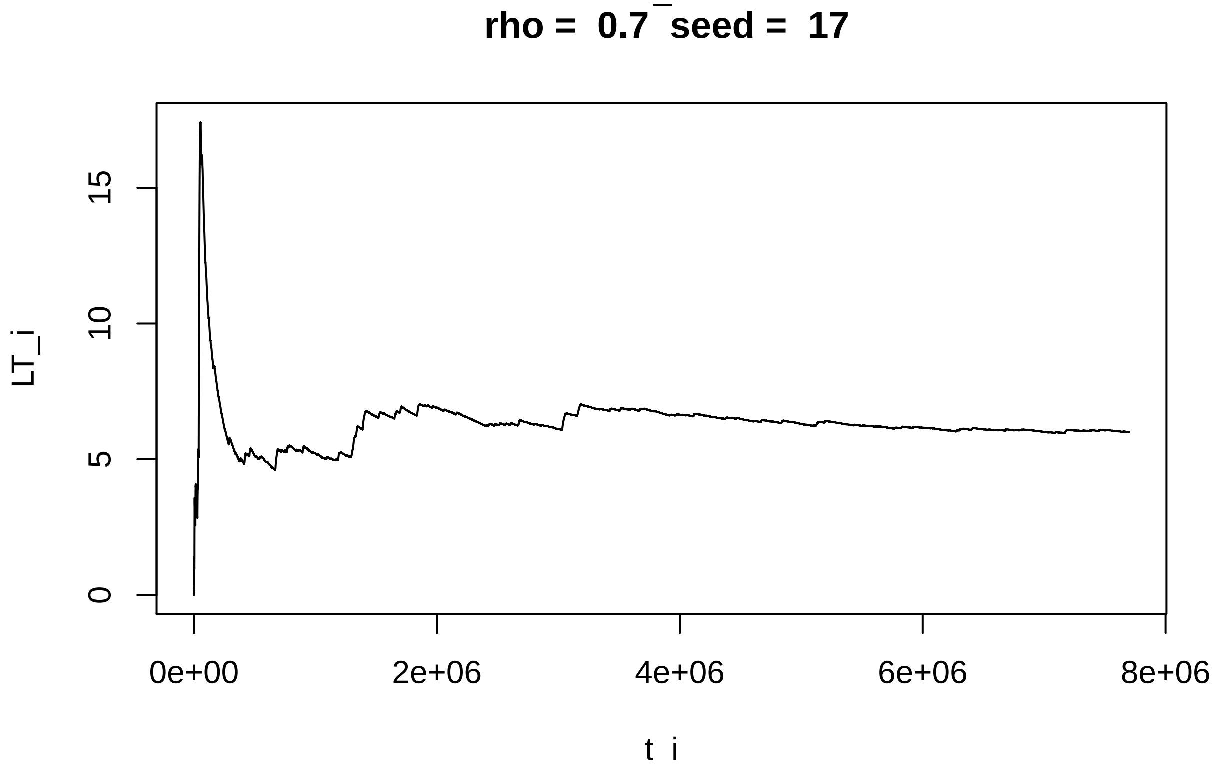

rho = 0.7 seed = 999
LT_i
t_i
0e+00
2e+06
4e+06
6e+06
8e+06
0
1
2
3
4
5
6


rho = 0.7 seed = 48
LT_i
t_i
0e+00
2e+06
4e+06
6e+06
8e+06
0
1
2
3
4
5
6

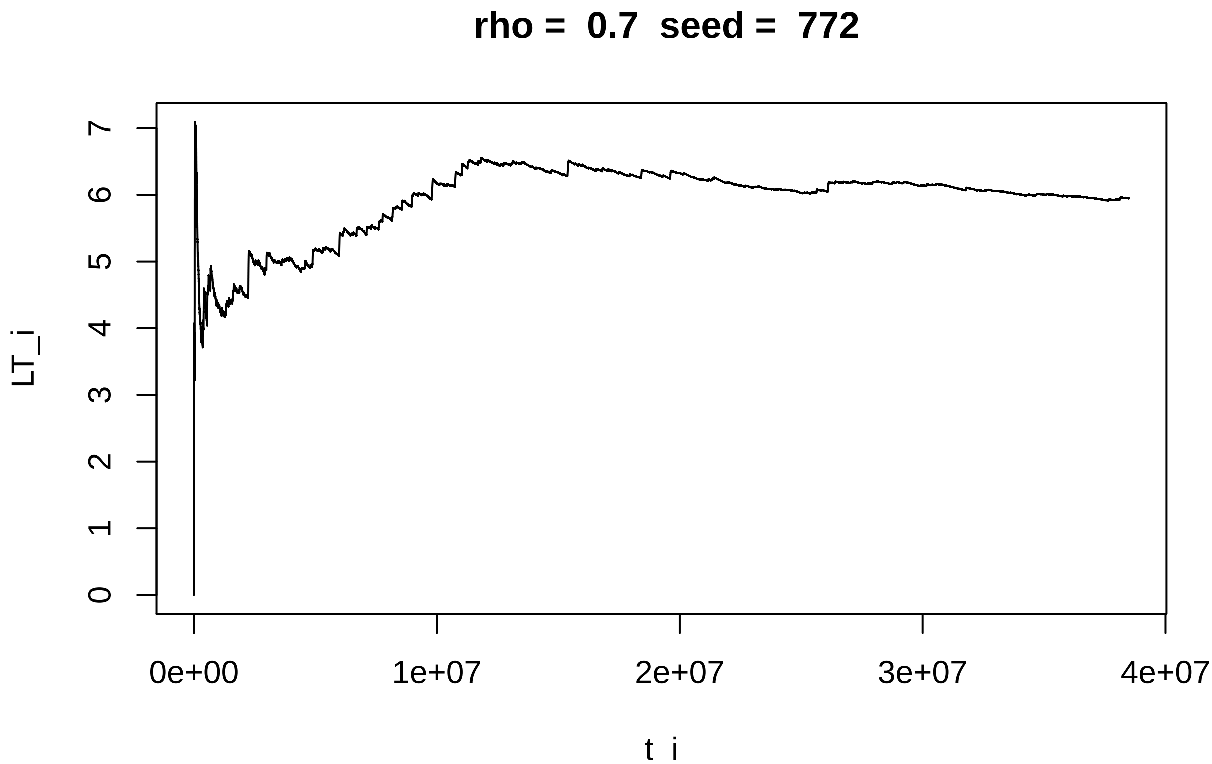

rho = 0.7 seed = 89
LT_i
t_i
0
2
4
6
8
10
12
0e+00
2e+06
4e+06
6e+06
8e+06

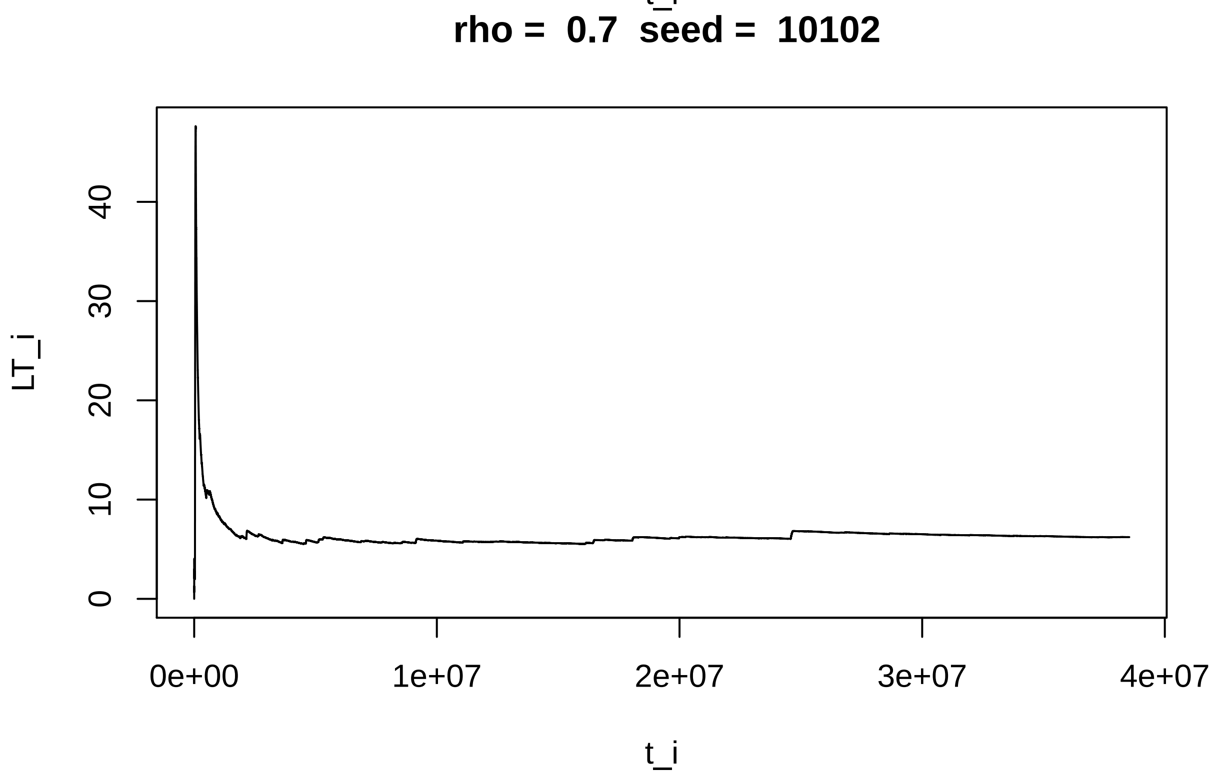

rho = 0.7 seed = 2001
LT_i
t_i
0
2
4
6
0e+00
2e+06
4e+06
6e+06
8e+06

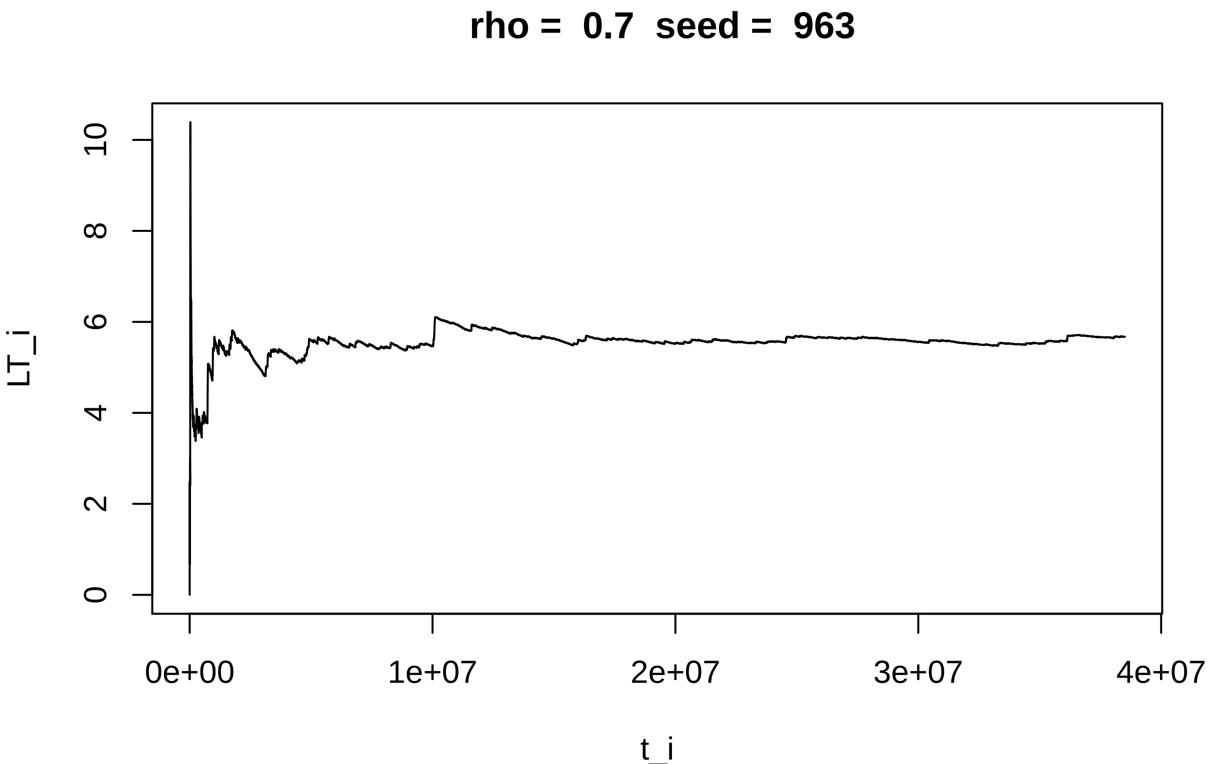


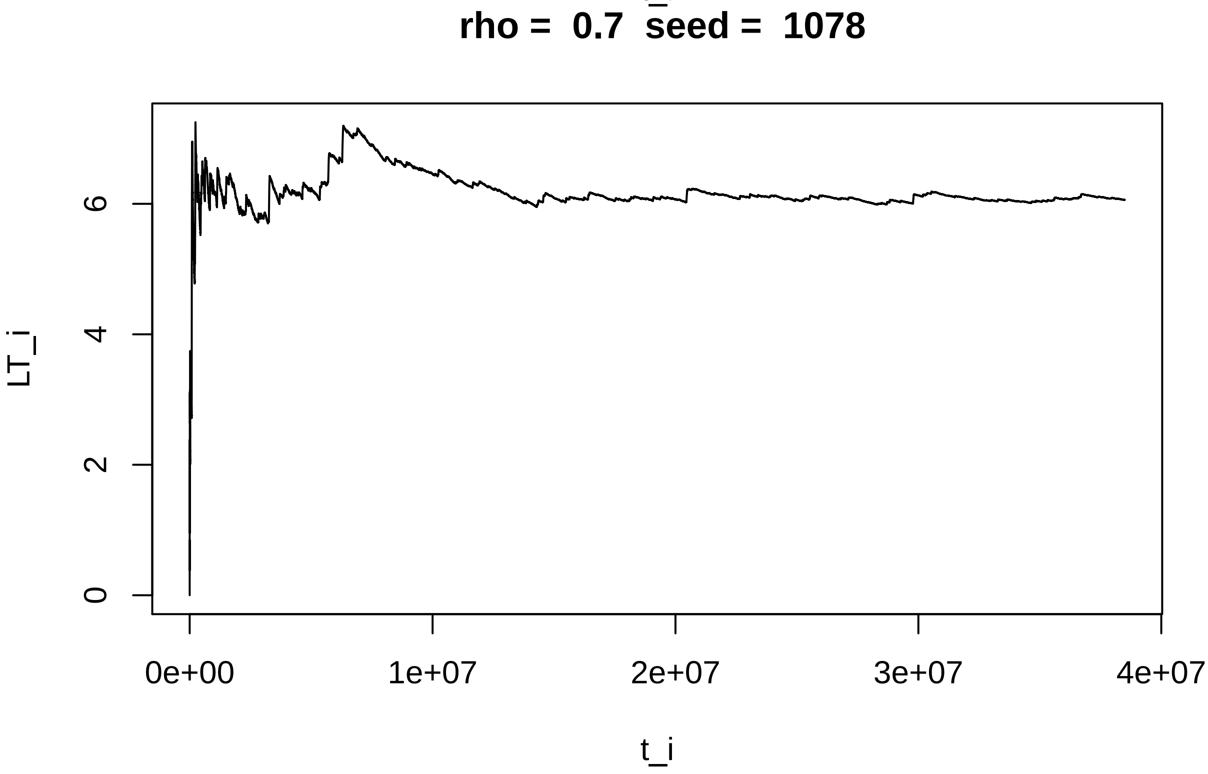


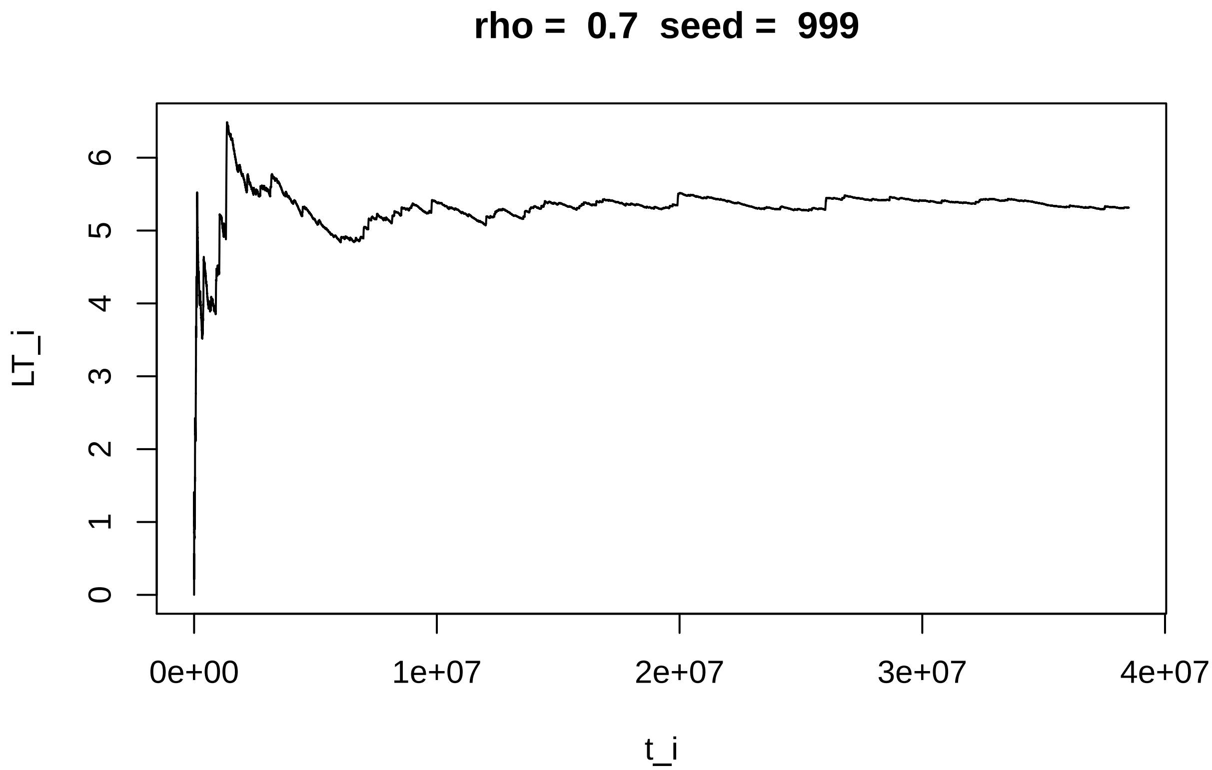
We observe that 5 out of 10 simulations are not in a steady state at the end of the simulation. We increase the number of clients to 500000 and repeat the simulations.

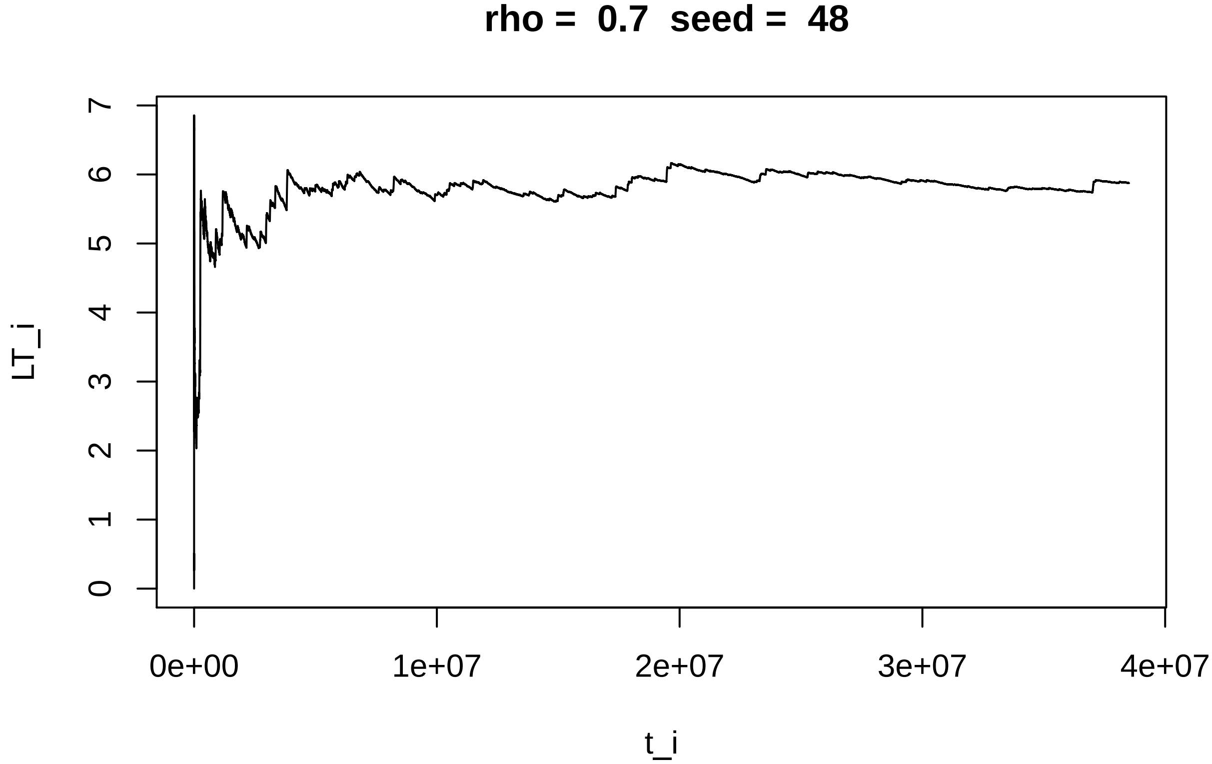

rho = 0.7 seed = 772
LT_i
0
1
2
3
4
5
6
7
0e+00
1e+07
2e+07
3e+07
4e+07
t_i

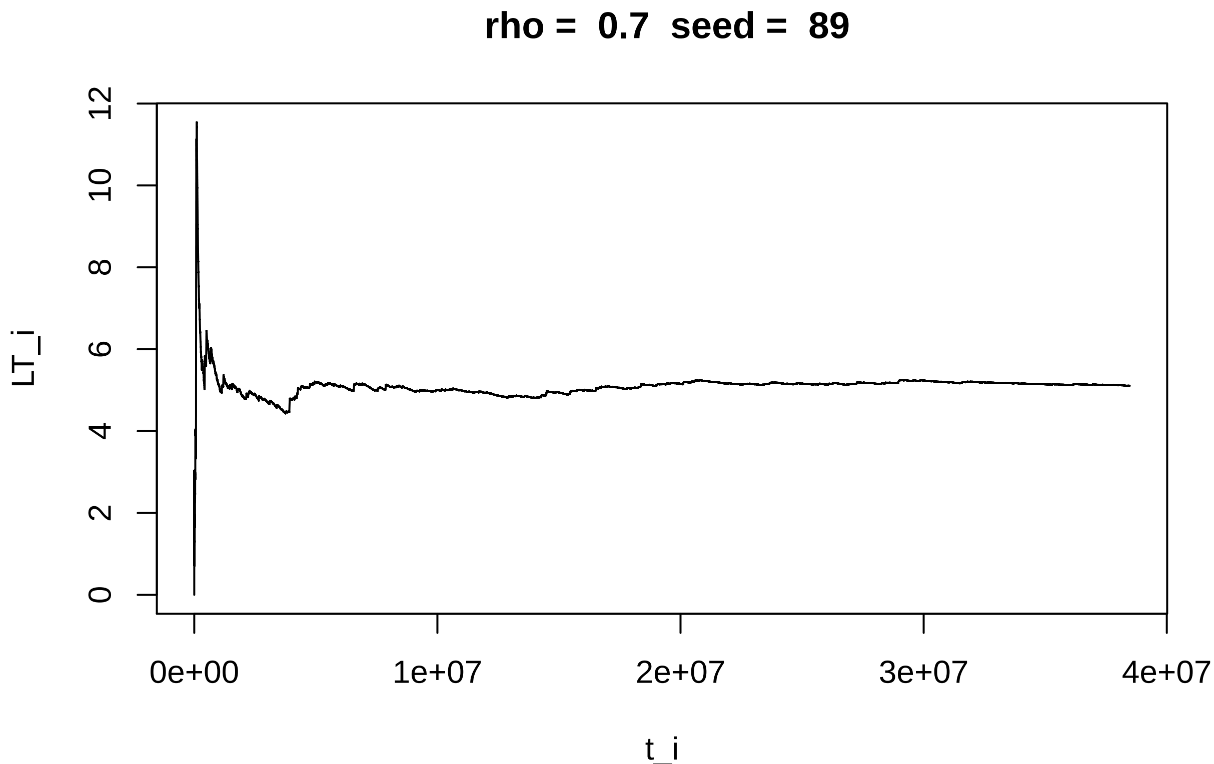

rho = 0.7 seed = 10102
LT_i
0
10
20
30
40
0e+00
1e+07
2e+07
3e+07
4e+07
t_i

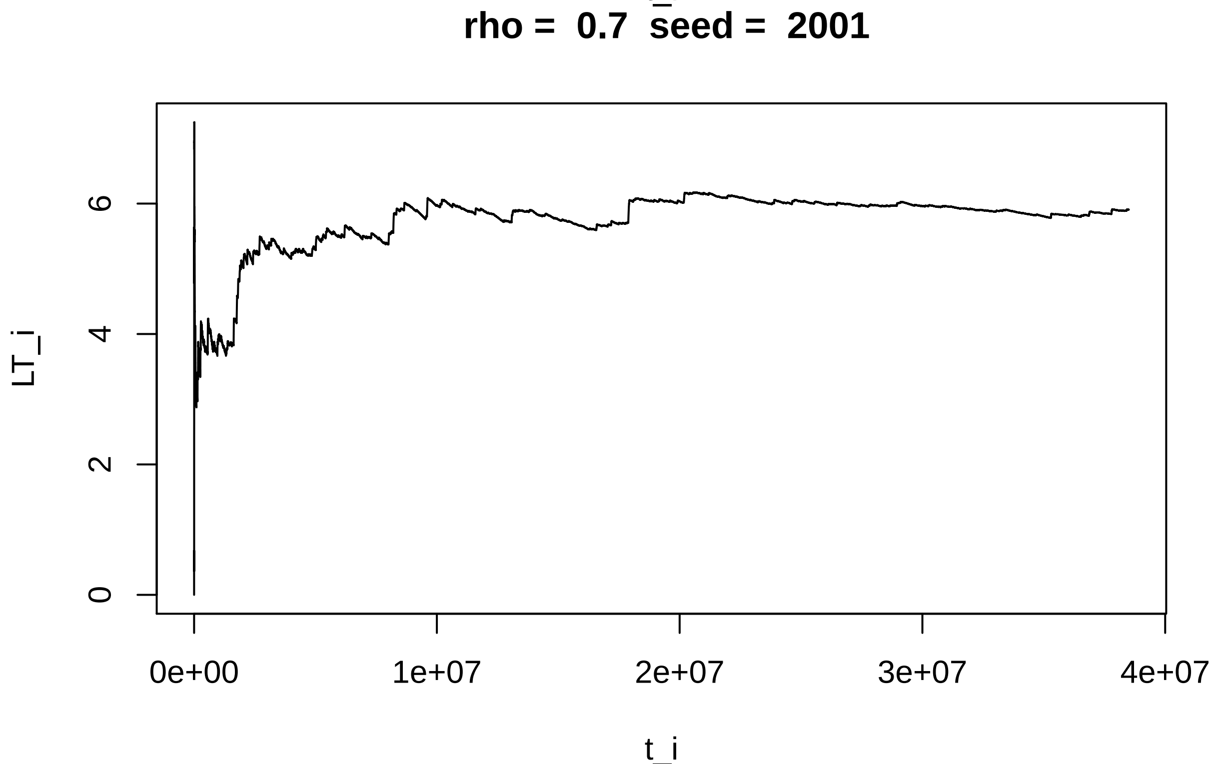

rho = 0.7 seed = 963
LT_i
t_i
0
2
4
6
8
10
0e+00
1e+07
2e+07
3e+07
4e+07

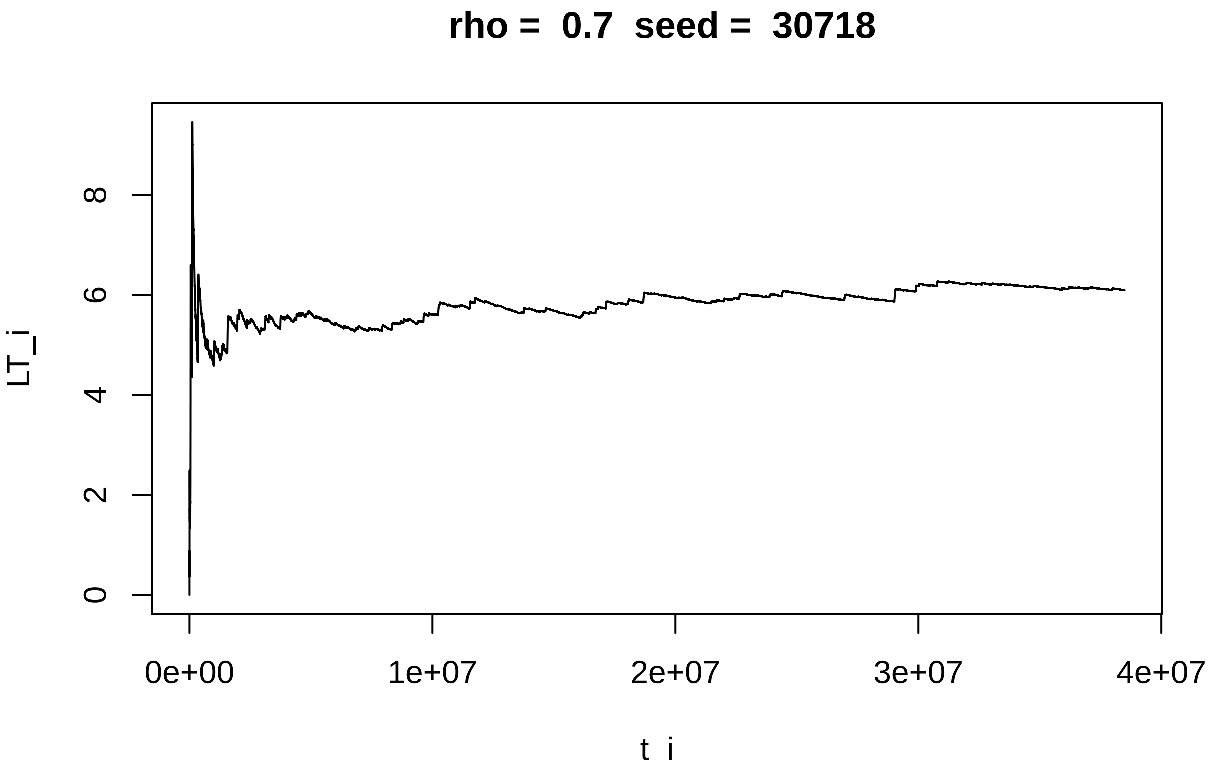

rho = 0.7 seed = 1078
LT_i
t_i
0
2
4
6
0e+00
1e+07
2e+07
3e+07
4e+07

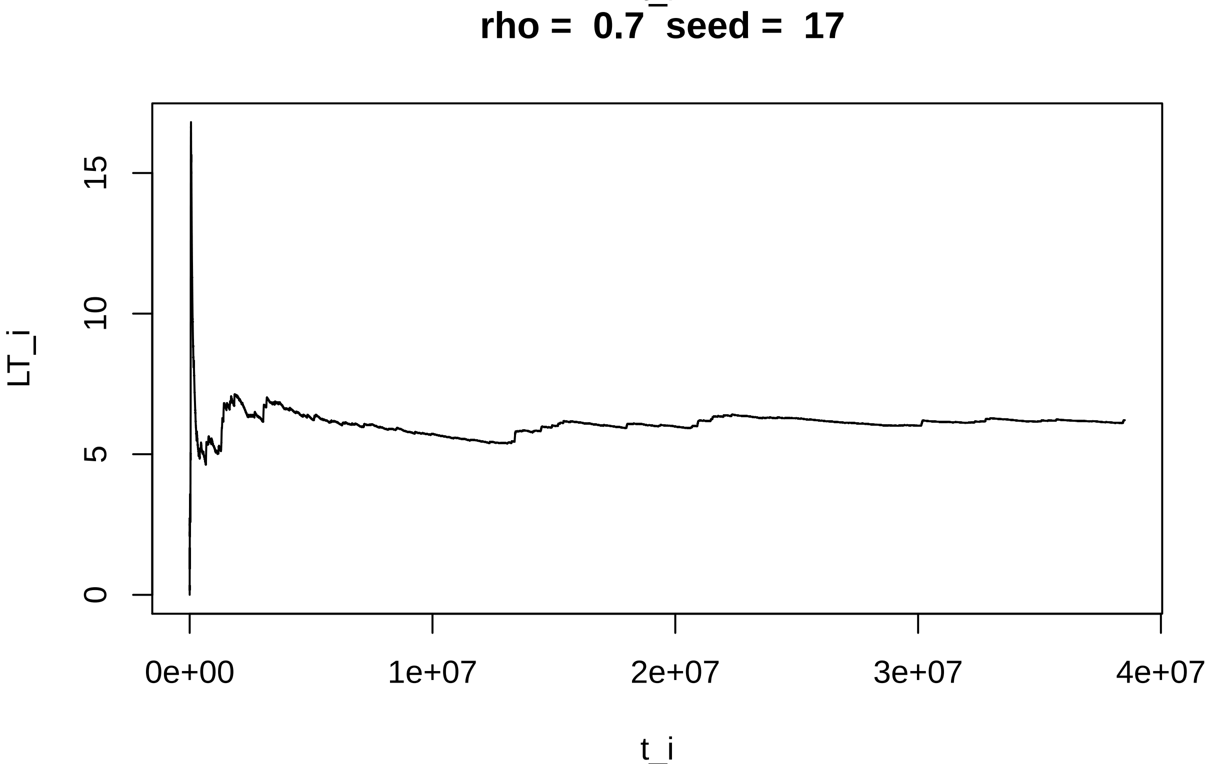

rho = 0.7 seed = 999
LT_i
0
1
2
3
4
5
6
0e+00
1e+07
2e+07
3e+07
4e+07
t_i


rho = 0.7 seed = 48
LT_i
0
1
2
3
4
5
6
7
0e+00
1e+07
2e+07
3e+07
4e+07
t_i


rho = 0.7 seed = 89
LT_i
0
2
4
6
8
10
12
0e+00
1e+07
2e+07
3e+07
4e+07
t_i


rho = 0.7 seed = 2001
LT_i
0
2
4
6
0e+00
1e+07
2e+07
3e+07
4e+07
t_i





We now compute the confidence interval for L_q and W_q using the same procedure we detailed for $\rho = 0.4$. We will use a t-Student distribution with critical value 1 - 0.95 and 9 degrees of freedom to compute a 95% confidence interval. The computations produce the followings confidence intervals for the average queue length and waiting time:

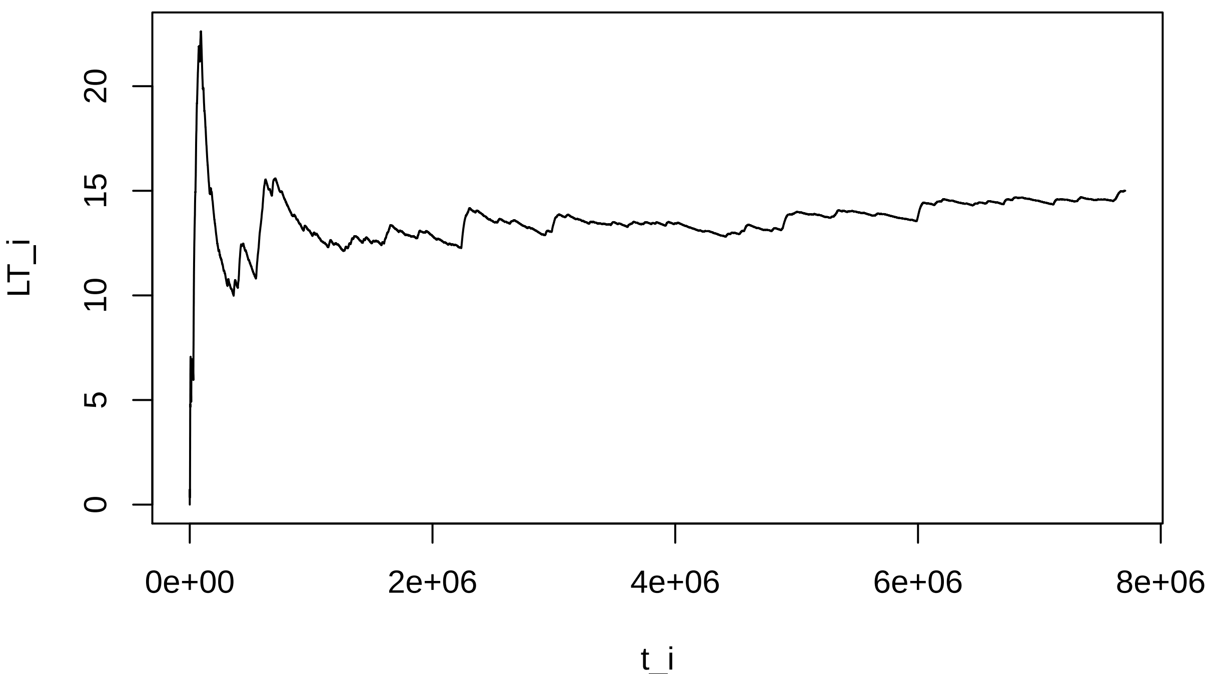
```
##    NA       NA       NA       NA       NA
```

```
## 2 0.7 4.926668 5.356066 379.3389 412.4915
```

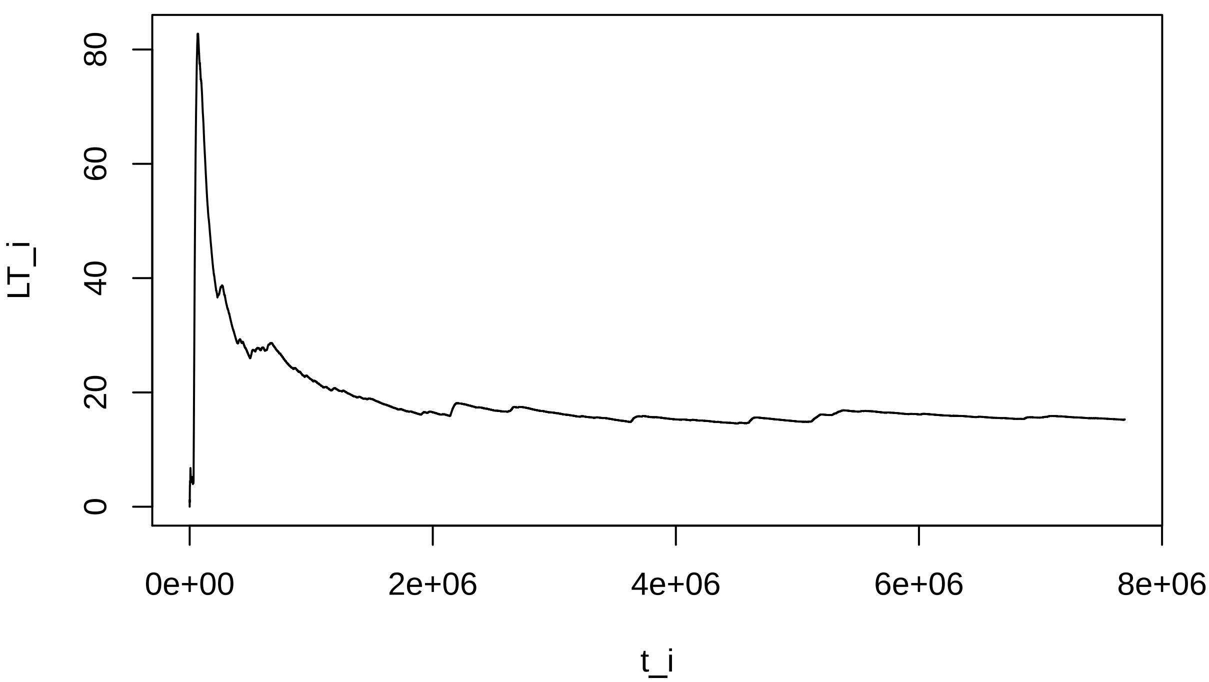
$\rho = 0.85$

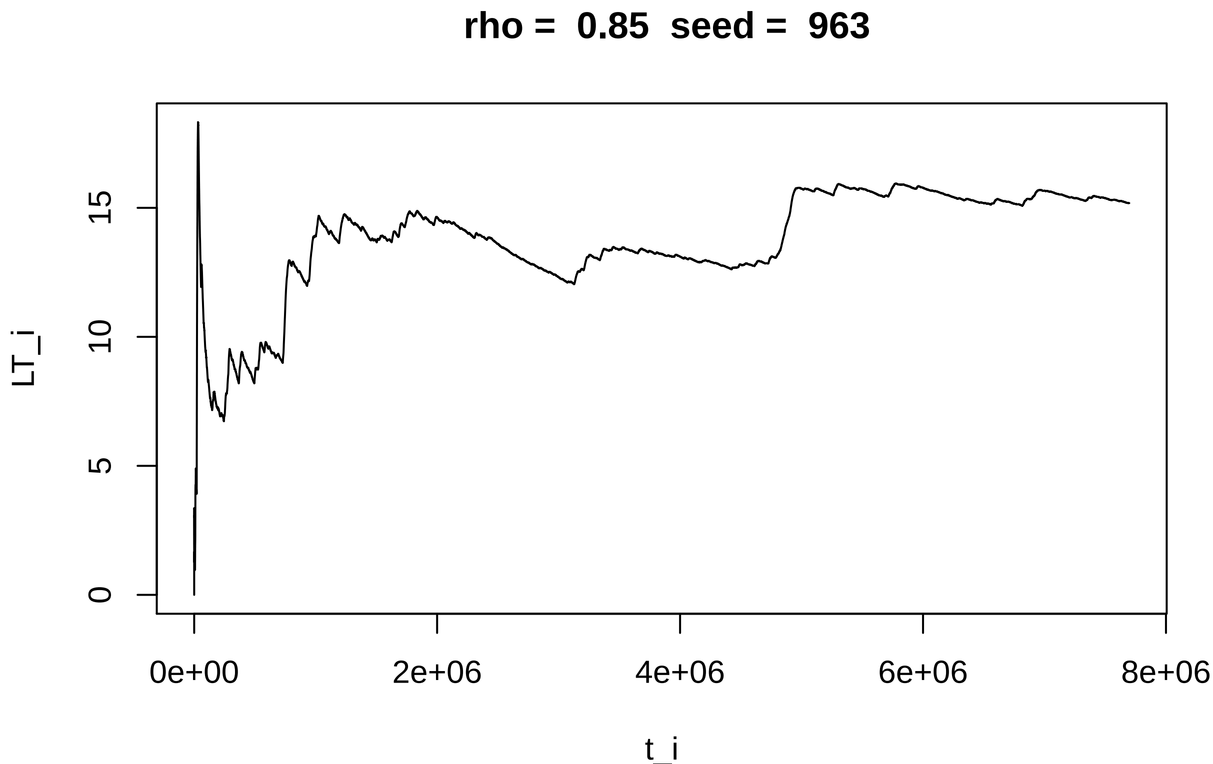
We generate 10 simulations with 100000 clients, each with a different seed. We check if the steady state is attained. We also check for irregularities in the simulation results, in which case we change the random number generator seed and repeat the simulation.

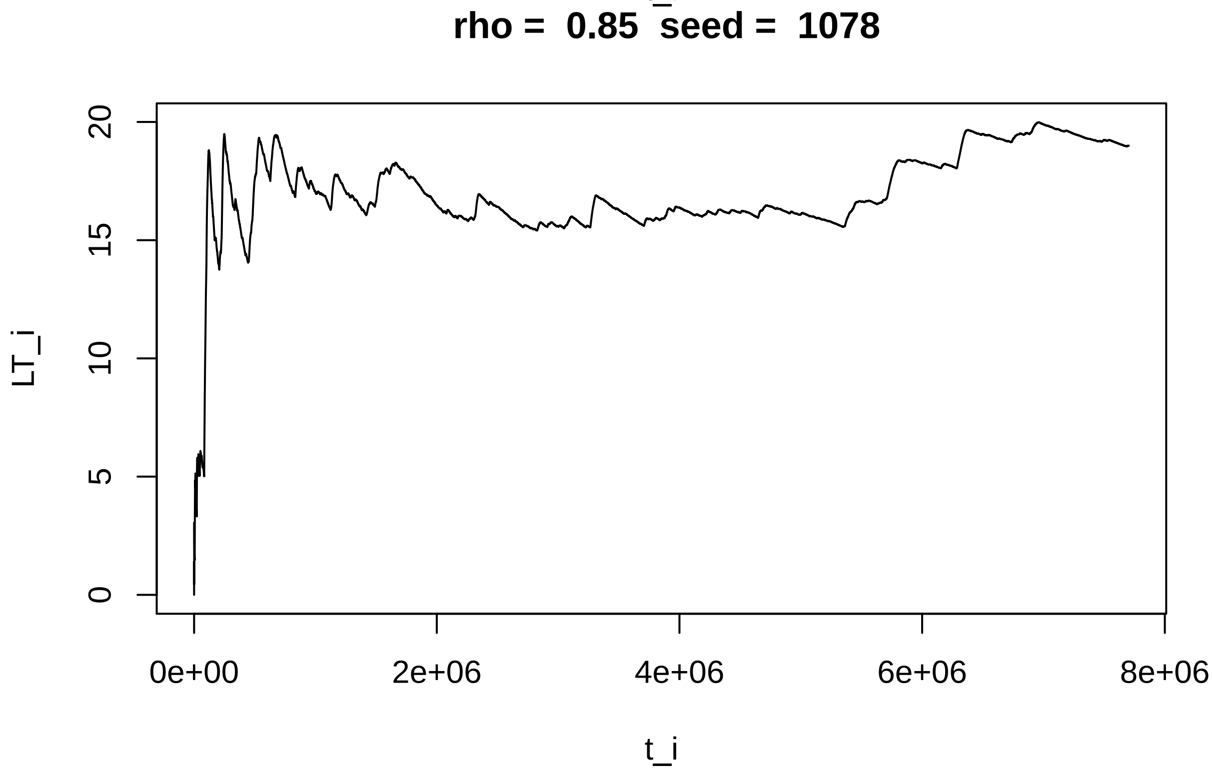
rho = 0.85 seed = 772

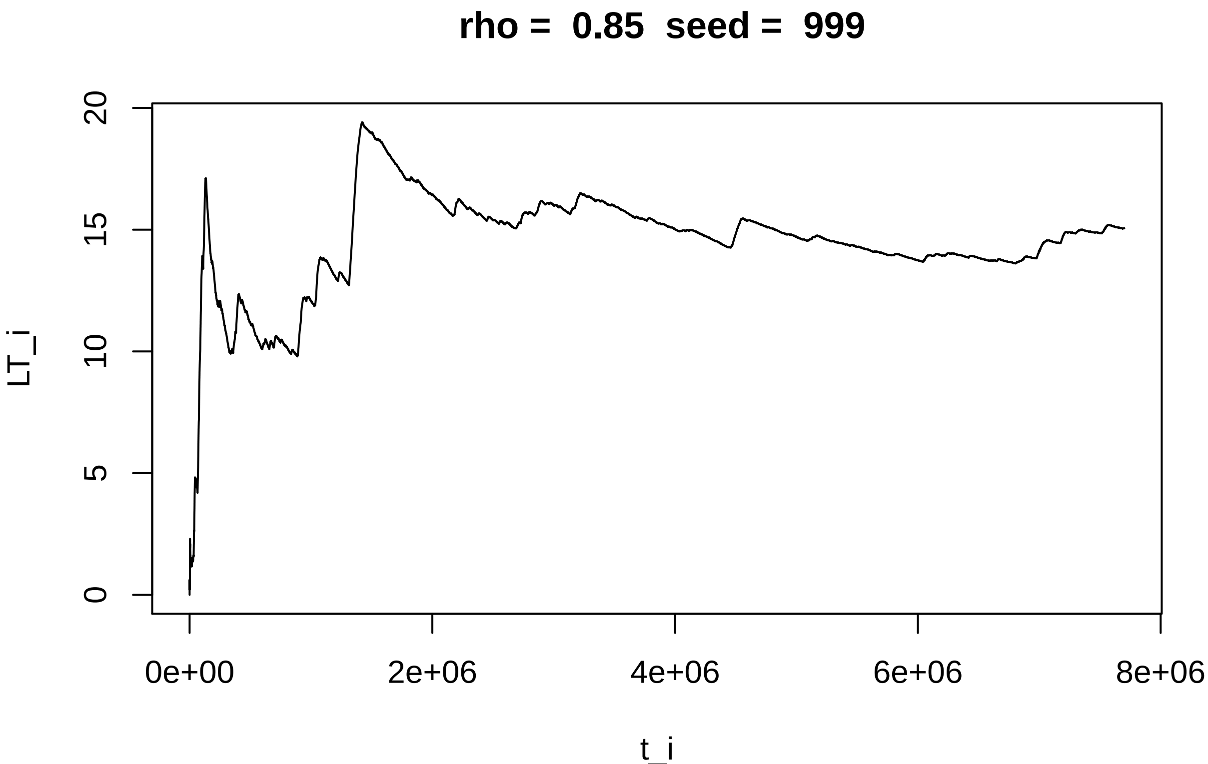


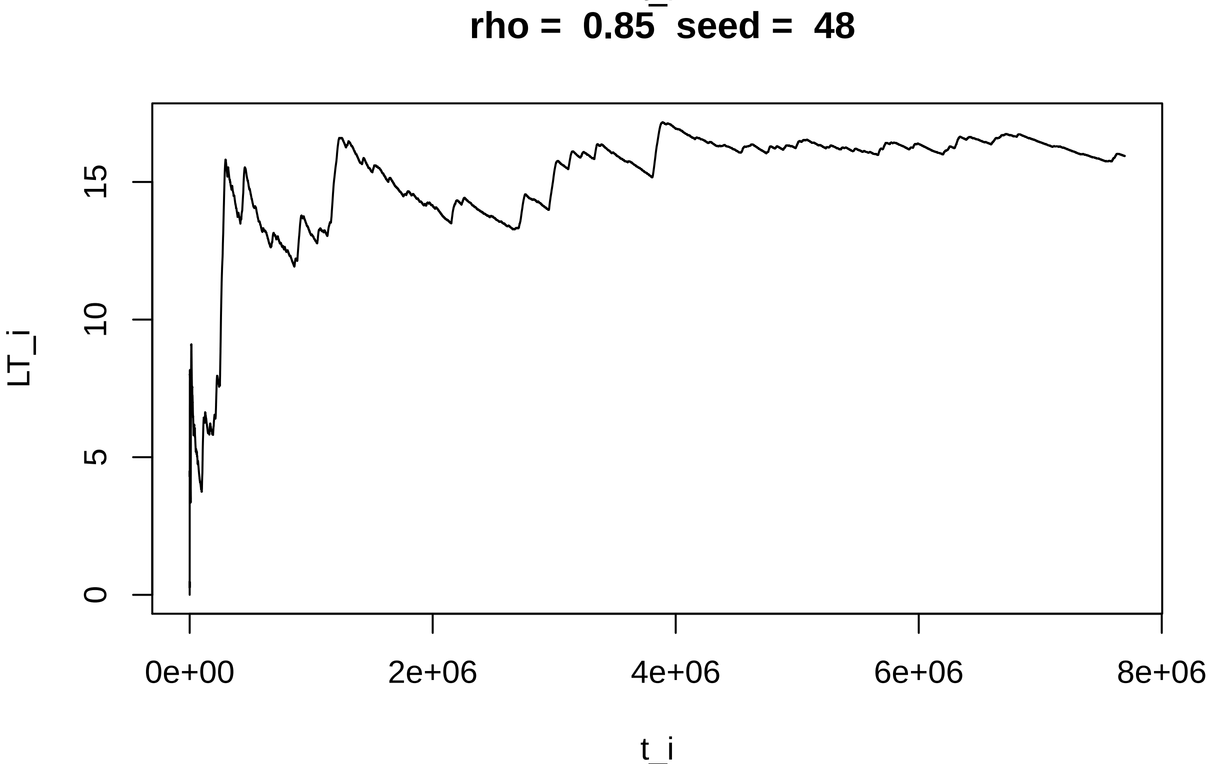
rho = 0.85 seed = 10102

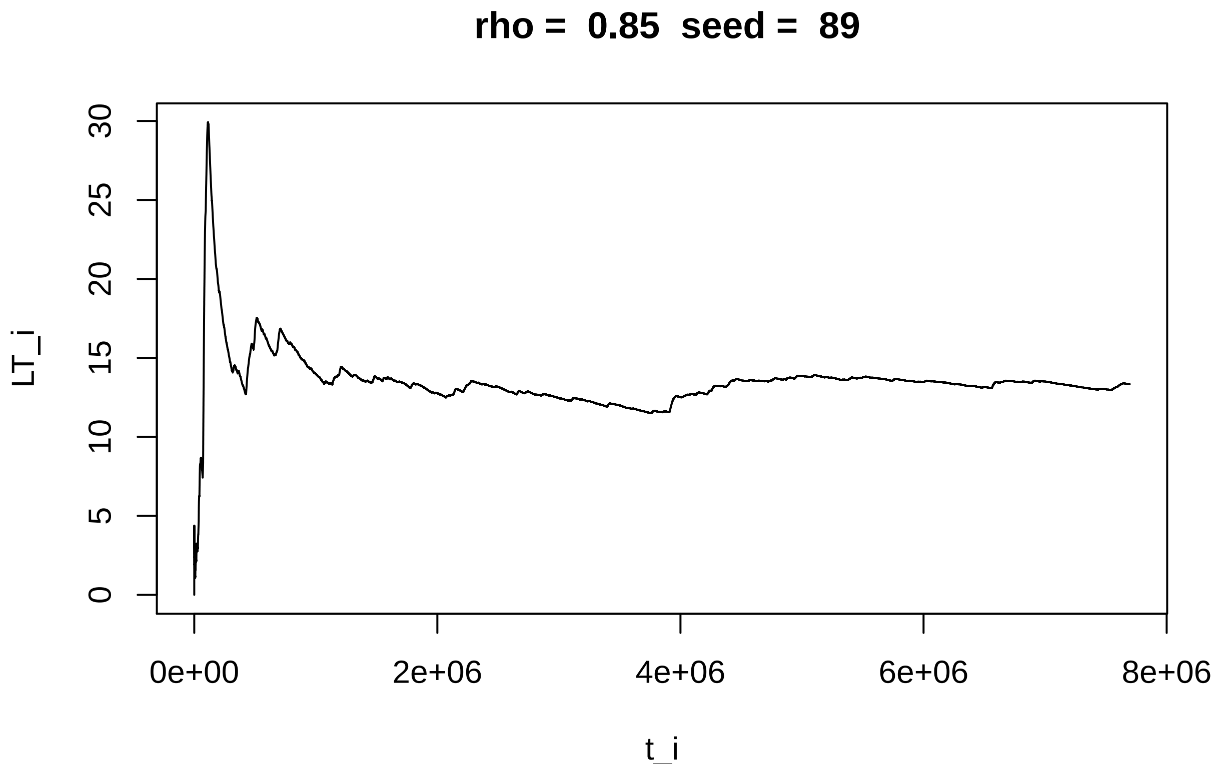


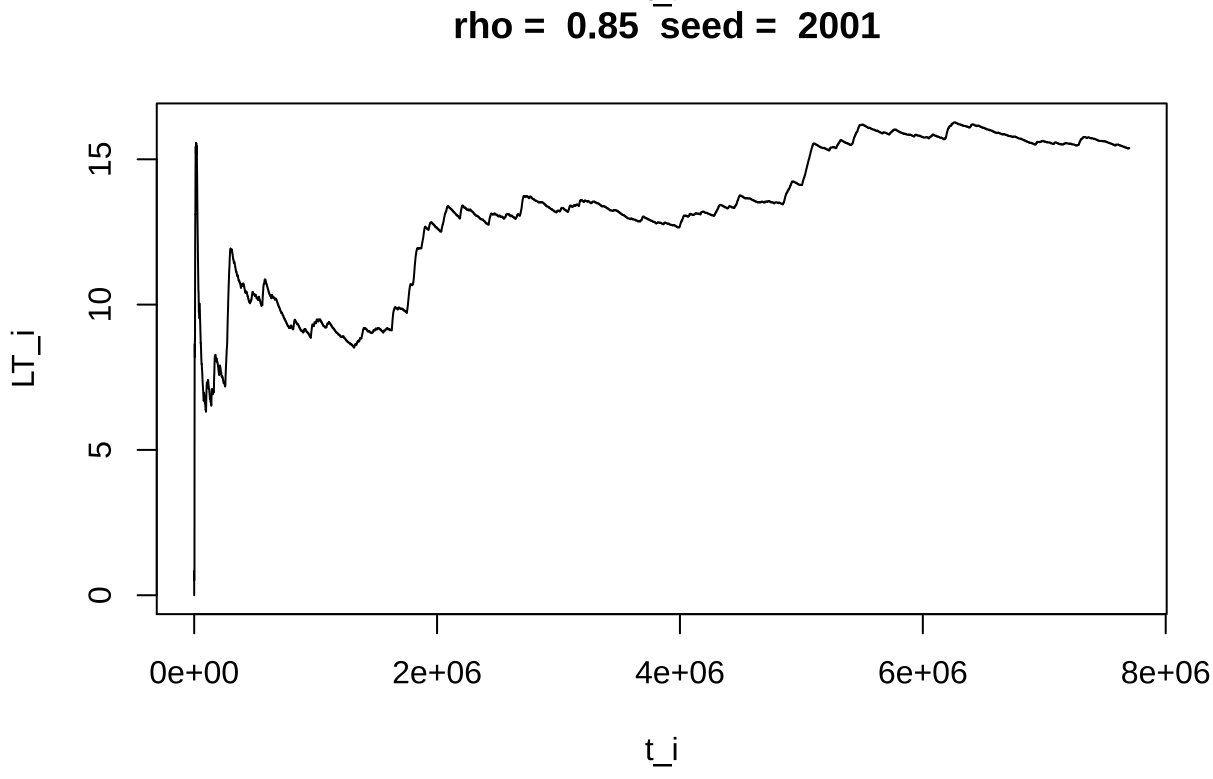

rho = 0.85 seed = 963
LT_i
0
5
10
15
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

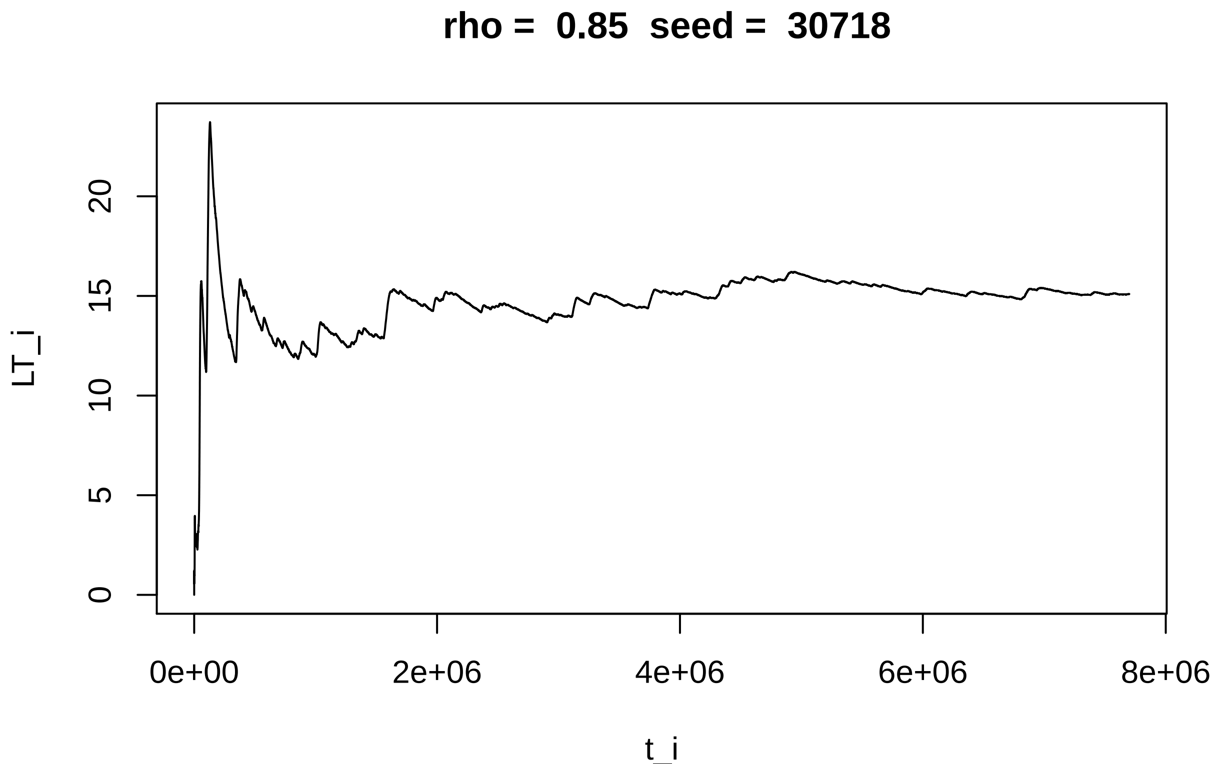

rho = 0.85 seed = 1078
LT_i
0
5
10
15
20
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

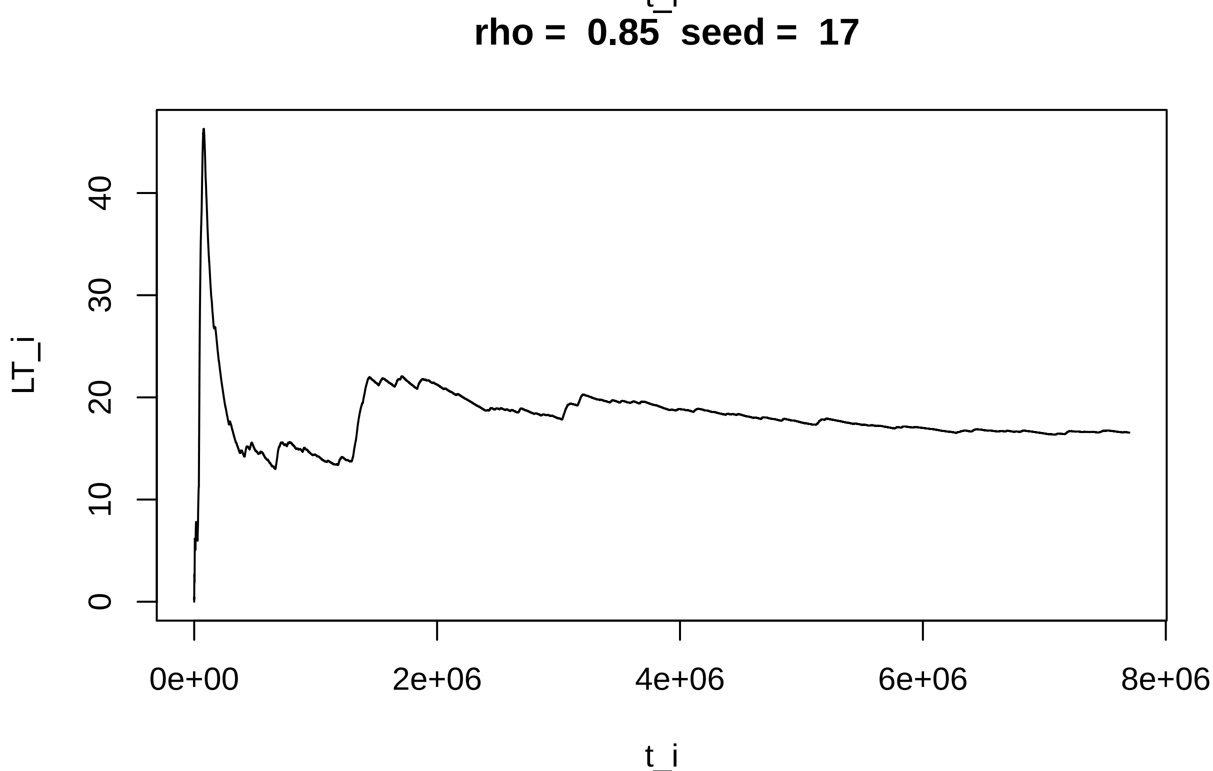

rho = 0.85 seed = 999
LT_i
0
5
10
15
20
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.85 seed = 48
LT_i
0
5
10
15
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.85 seed = 89
LT_i
0
5
10
15
20
25
30
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.85 seed = 2001
LT_i
0
5
10
15
0e+00
2e+06
4e+06
6e+06
8e+06
t_i





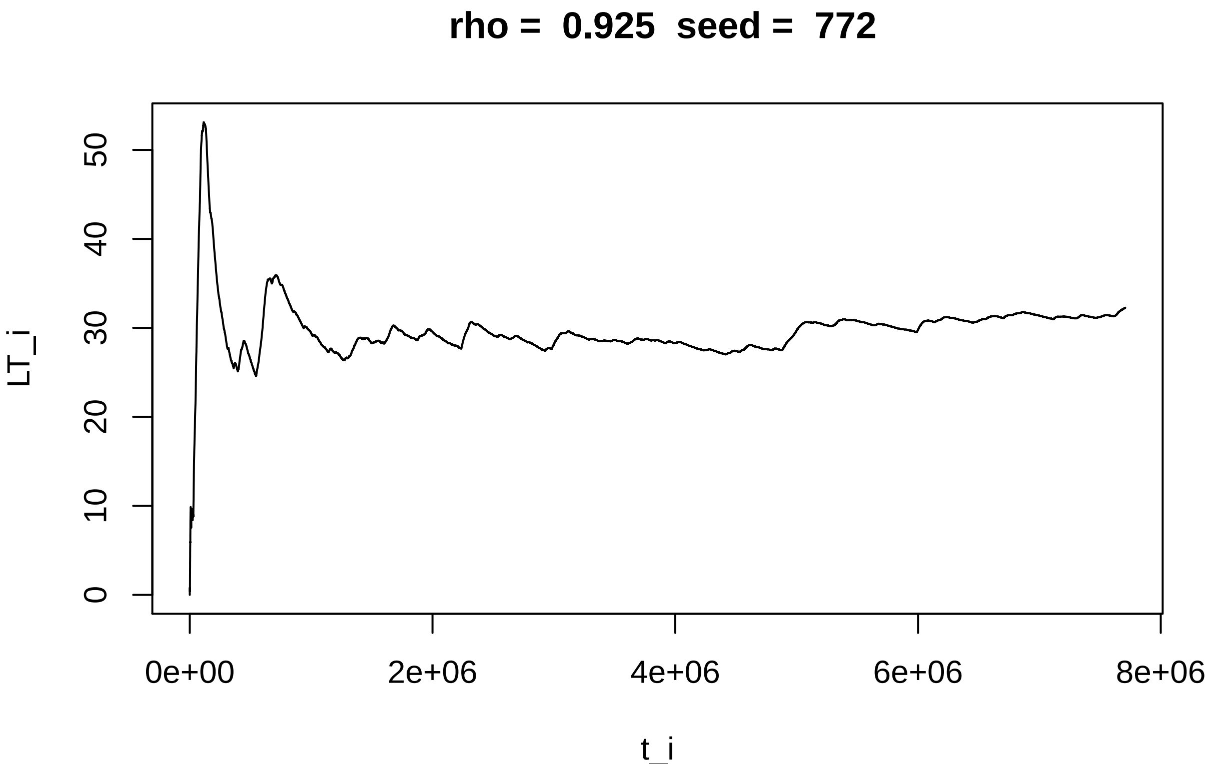
We observe that ... PENDING

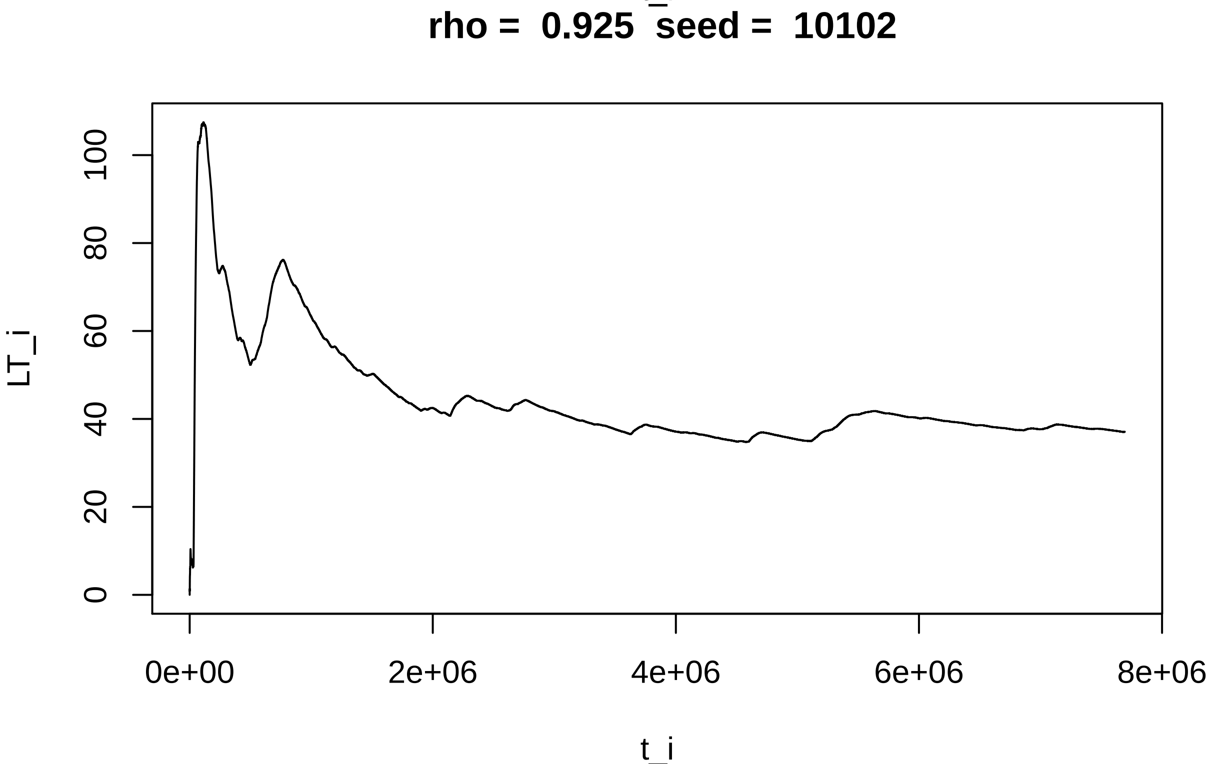
We now compute the confidence interval for L_q and W_q using the same procedure we detailed for $\rho = 0.4$. We will use a t-Student distribution with critical value 1 - 0.95 and 9 degrees of freedom to compute a 95% confidence interval. The computations produce the followings confidence intervals for the average queue length and waiting time:

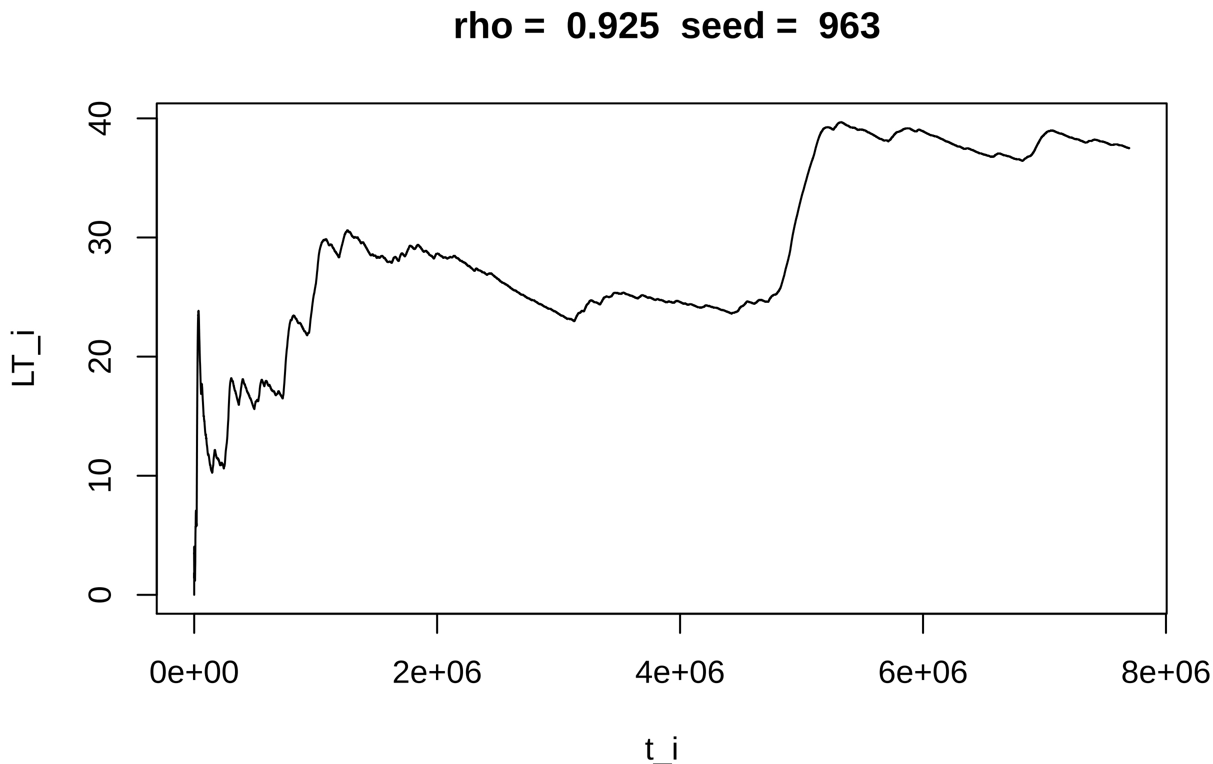
```
##     NA       NA       NA       NA       NA
## 3 0.85 13.89277 15.56913 1069.562 1198.859
```

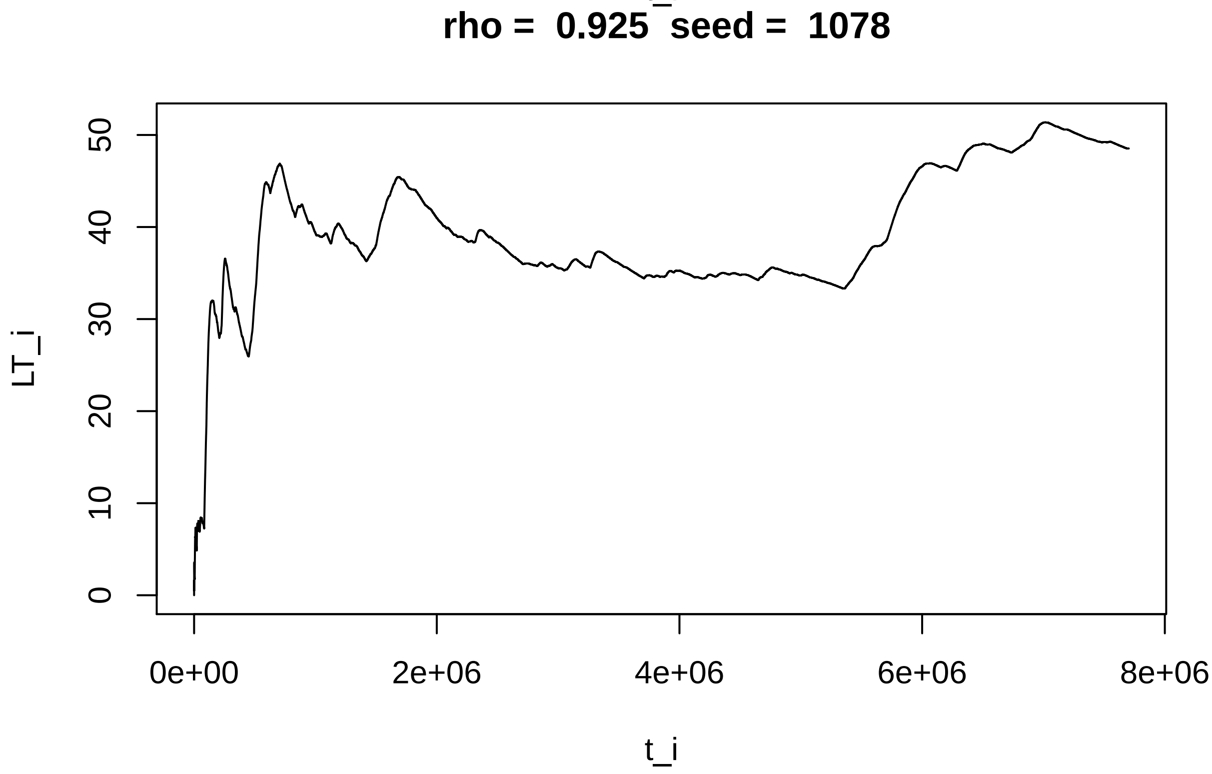
$\rho = 0.925$

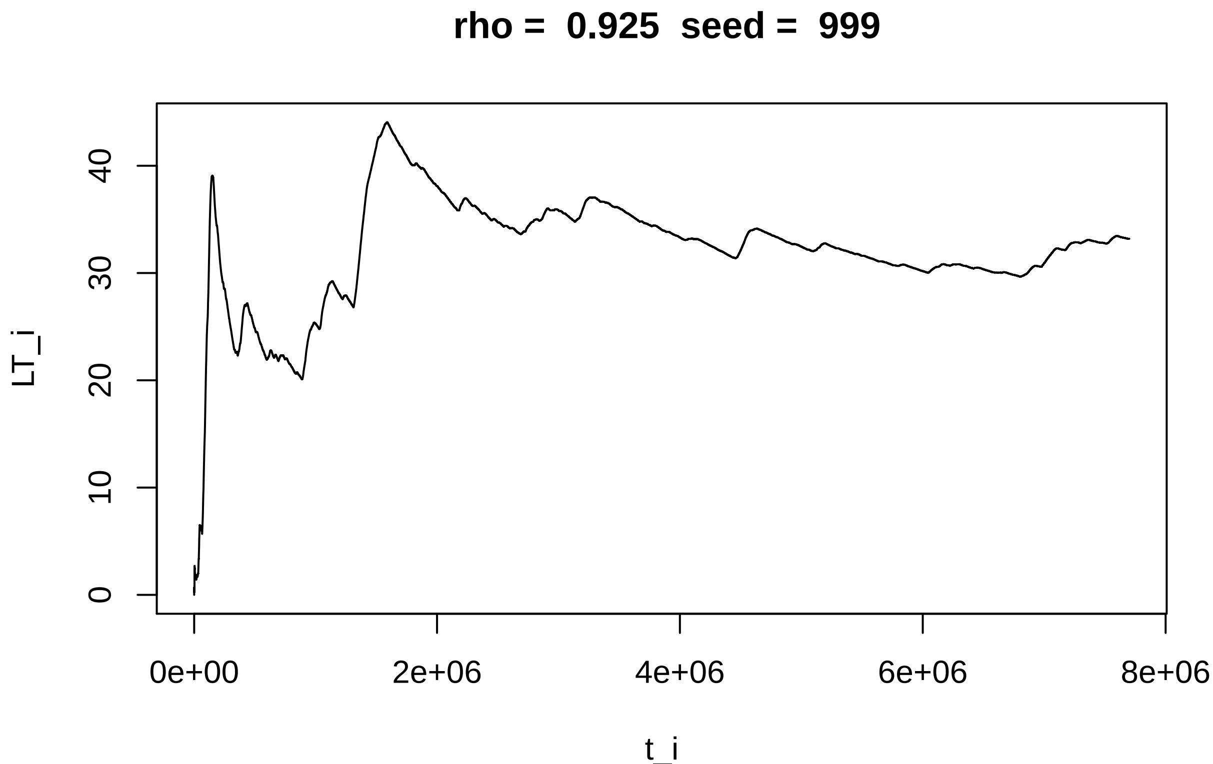
We generate 10 simulations with 100000 clients, each with a different seed. We check if the steady state is attained. We also check for irregularities in the simulation results, in which case we change the random number generator seed and repeat the simulation.

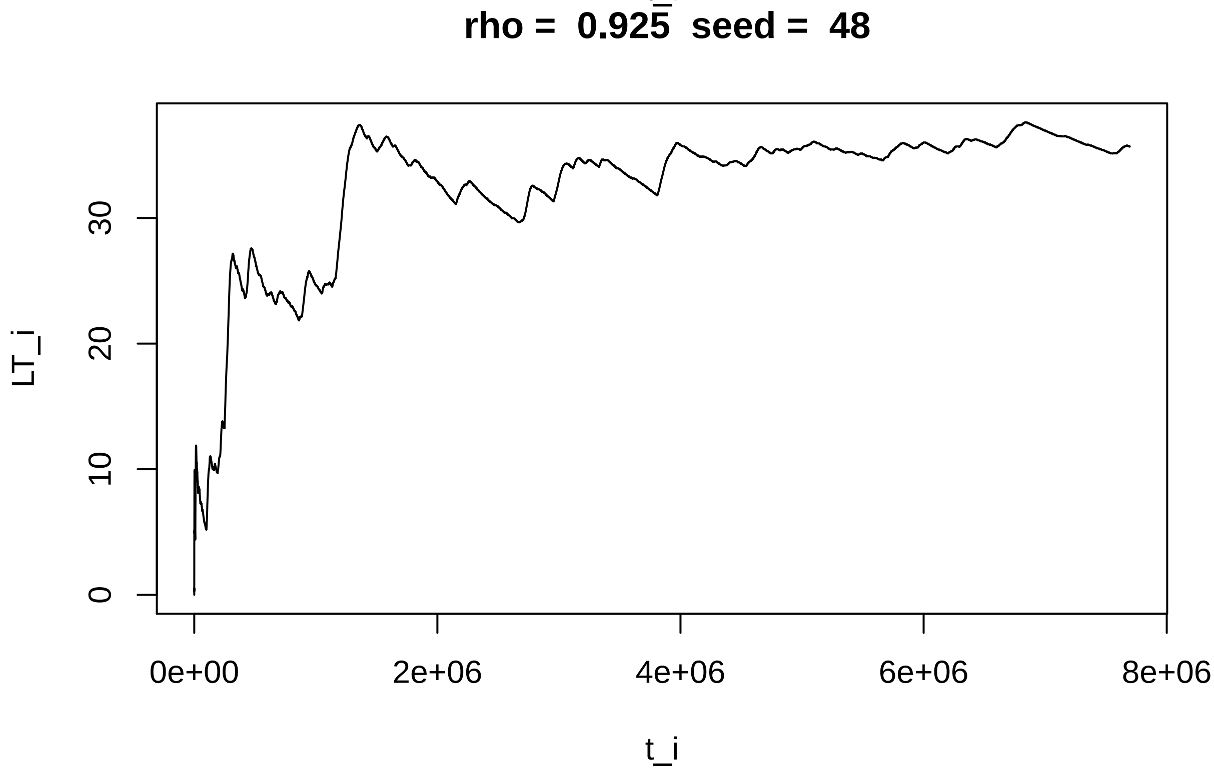

rho = 0.925 seed = 772
LT_i
0
10
20
30
40
50
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

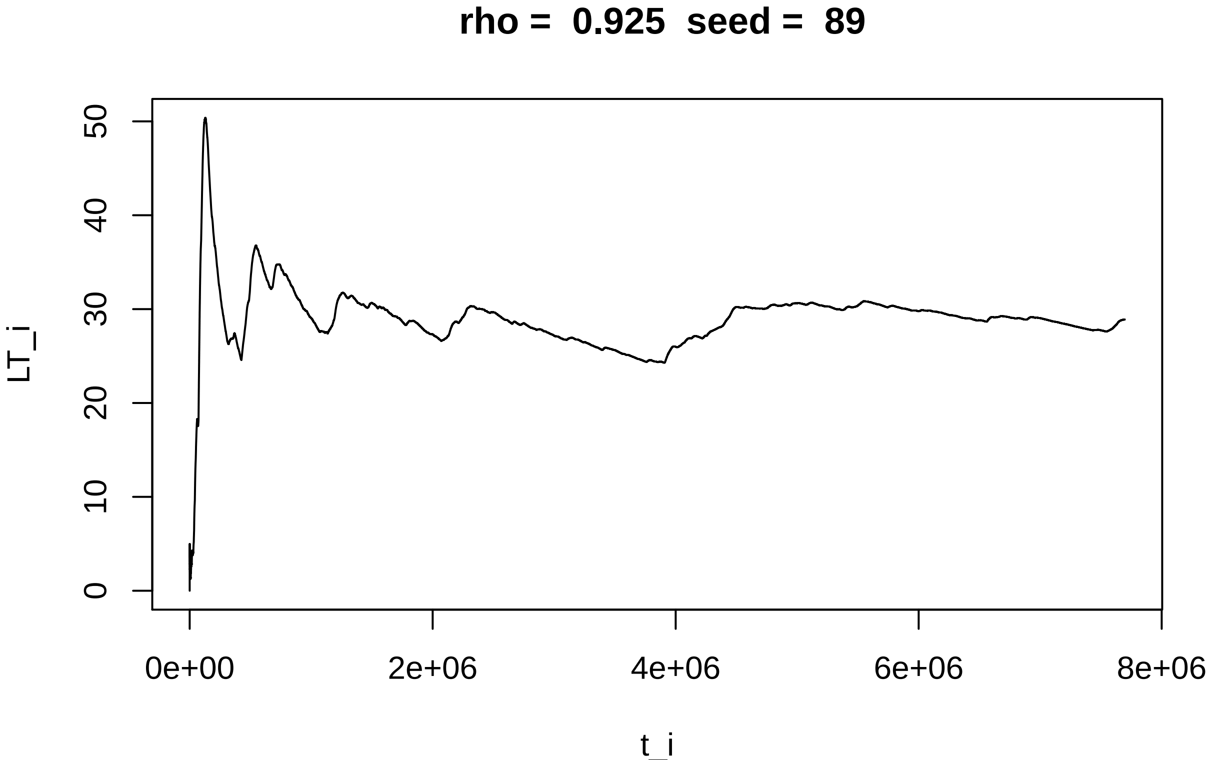

rho = 0.925 seed = 10102
LT_i
0
20
40
60
80
100
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

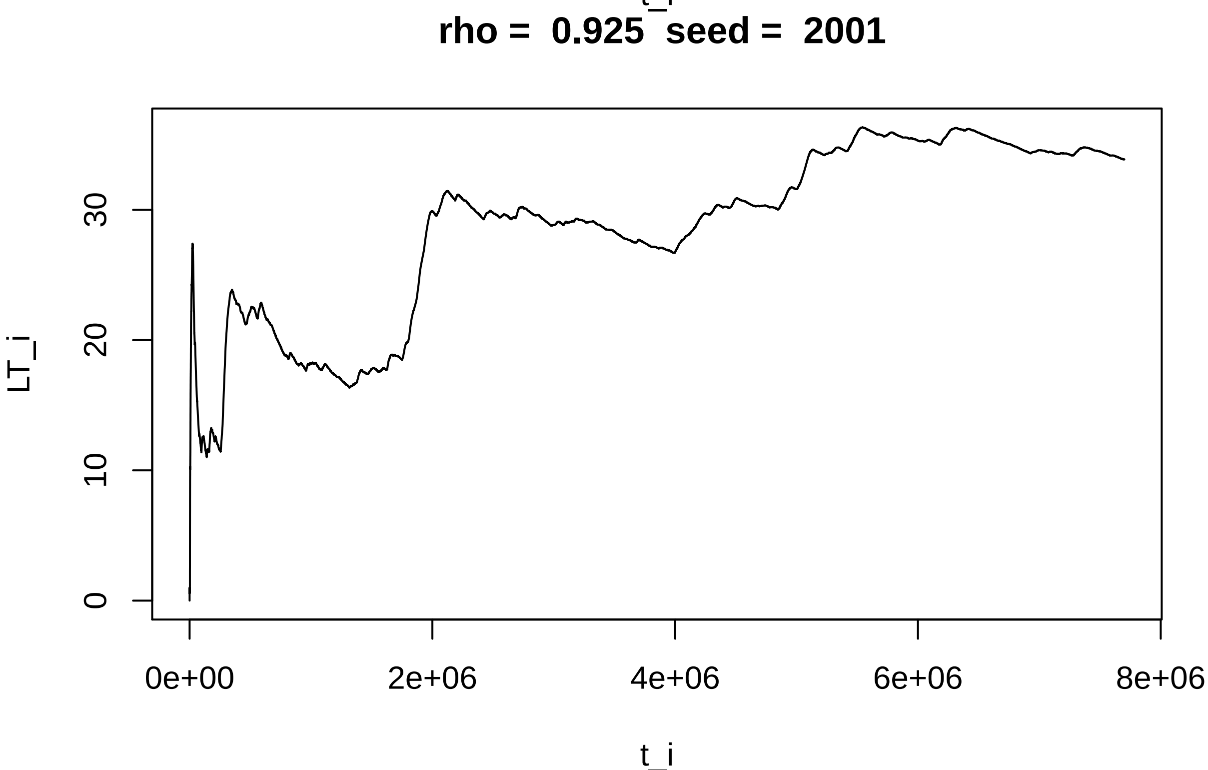

rho = 0.925 seed = 963
LT_i
0
10
20
30
40
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

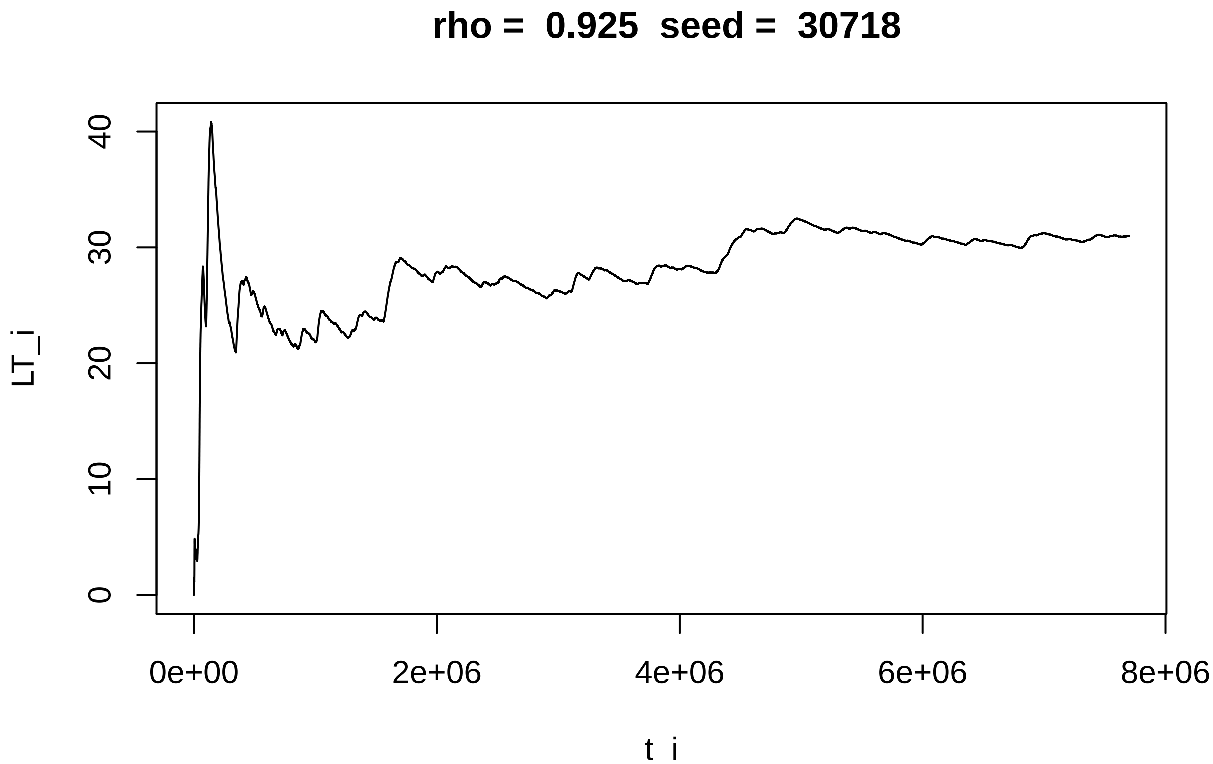

rho = 0.925 seed = 1078
LT_i
0
10
20
30
40
50
0e+00
2e+06
4e+06
6e+06
8e+06
t_i

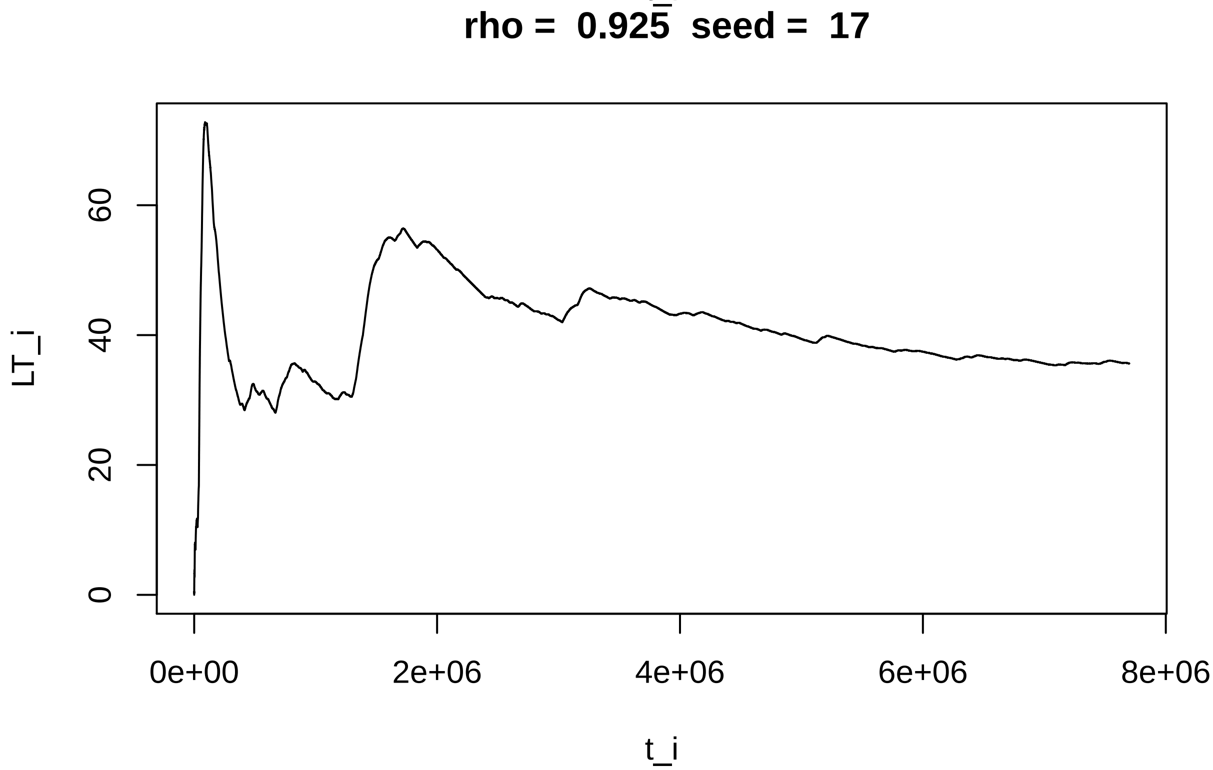

rho = 0.925 seed = 999
LT_i
0
10
20
30
40
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.925 seed = 48
LT_i
0
10
20
30
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.925 seed = 89
LT_i
0
10
20
30
40
50
0e+00
2e+06
4e+06
6e+06
8e+06
t_i


rho = 0.925 seed = 2001
LT_i
0
10
20
30
0e+00
2e+06
4e+06
6e+06
8e+06
t_i





We observe that . . . PENDING

We now compute the confidence interval for L_q and W_q using the same procedure we detailed for $\rho = 0.4$. We will use a t-Student distribution with critical value 1 - 0.95 and 9 degrees of freedom to compute a 95% confidence interval. The computations produce the followings confidence intervals for the average queue length and waiting time:

```
##      NA       NA      NA       NA       NA
## 4 0.925 31.32943 37.5434 2412.028 2890.821
```

Comparison of Allen Cuneen's approximation and the simulation

ρ	W_q	-C.I W_q	+C.I W_q	L_q	-C.I L_q	+C.I L_q
0.4	69.1507968	51.0353731	55.5687961	0.8980623	0.6629266	0.7216626
0.7	423.5486306	379.3389492	412.4915102	5.5006316	4.9266678	5.3560657
0.85	1249.0362678	1069.5617305	1198.8594895	16.2212502	13.8927651	15.5691295
0.925	2958.3575269	2412.0275501	2890.8210733	38.4202276	31.3294274	37.5433959