SPTF SCHEDULING PROJECT

Sommario

[TRACK 3](#_Toc125811223)

[Considerations 3](#_Toc125811224)

[1 – Implement an M/G/1 system 4](#_Toc125811225)

[2 – omnetpp.ini file 4](#_Toc125811226)

[3 – Statistics collection 4](#_Toc125811227)

[3.1 – Average queueing time 4](#_Toc125811228)

[3.2 – Average queueing time conditioned to packet length 4](#_Toc125811229)

[3.3 – Average response time 5](#_Toc125811230)

[3.4 – Utilization factor of the server 5](#_Toc125811231)

[3.5 – Queue length over time 5](#_Toc125811232)

[4 – Compare experimental and theoretical values 5](#_Toc125811233)

[4.1 – Average conditional queue time 5](#_Toc125811234)

[4.2 – Average response time 5](#_Toc125811235)

[4.3 – Utilization factor of the server 5](#_Toc125811236)

# TRACK

1. Implement a M/G/1 system with a SPTF discipline where G is Uniform(0,L);
2. All input parameters must be NED parameters in omnetpp.ini;
3. Collect statistics on:
   1. Average queueing time;
   2. Average queueing time conditioned to the packet’s length;
   3. Average response time;
   4. Utilization factor of the server;
   5. Queue length over time.
4. Compare experimental and theoretical values of:
   1. Average conditional queueing time;
   2. Average response time;
   3. Utilization factor of the server.

# Considerations

//

As expected from Kleinrock’s conservation law, assigning a priority class mean queuing time shrinks, indeed also mean response time (being summation of mean queuing time which shrinks and mean service time which is constant) is reduced. Using Little’s law yields that even average queue length shrinks (average queue length is equal to waiting time by a coefficient of lambda = arrival rate).

//La SPTF ha un ottimo vantaggio secondo me, che è quello di mantenere un altissimo numero di utenti serviti al minuto indipendentemente dalle condizioni di stabilità del sistema.

Tuttavia questa caratteristica non è ben visibile in condizioni di stabilità, che sono le condizioni in cui noi siamo in grado di calcolare qualunque cosa.

Per questo motivo sono state eseguite due simulazioni. D’ora in poi farò riferimento a NET2 e NET3 e saranno le due simulazioni eseguite con i seguenti parametri:

1. NET2: blah blah blah
2. NET3: blah blah bnlah

# 1 – Implement an M/G/1 system

We implemented an M/G/1 system in which service time G is distributed as an Uniform distribution between 0 and L.

The system has tree modules. Source, Queue and Sink.

Source generates messages with an exponential distribution with average λ, then it is assigned to each message a computation time distributed uniformly between 0 and L. This value is attached to a message by using the class Mail, which is an extension of the class cMessage.

Queue receive Mails and process it if server is empty. If it is not, mails are inserted into a cQueue. Right after the insertion of a new mail, the cQueue is sorted, to guarantee the SPTF scheduling. Surely this could be done in a more efficient way, but this was the easiest to implement and on a stable system it does not impact so much on the simulation.

Sink receive the packets and collects some statistics. Then packets are destroyed.

# 2 – omnetpp.ini file

In the file omnetpp.ini is possible to make a custom simulation by editing the parameters. Is it possible so edit the end of uniform distribution L, and the average of packets arrival λ.

It is also possible to change default simulation processing time and simulation waiting time.

# 3 – Statistics collection

Statistics and data are collected in each module by Omnet.

Omnet editor was enough for compute some statistics, but the simulation generates also a log.txt file with a format that fit well with Matlab importdata() function.

In particular, the log is useful to associate a packet with its length (process time) and it’s waiting time in the queue. This is necessary to compute packet queue time conditioned to their size.

All the following statistics refer to a simulation with the following values:

so that .

# 3.1 – Average queueing time

Average queueing time can be easily computed knowing λ and μ in a M/M/n system.

Average queueing time is computed with the Omnet tool, and it is:

# 3.2 – Average queueing time conditioned to packet length

As mentioned above, to compute the conditioned queue time, the simulation generates a log file, which is then computed with Matlab.

Matlab takes as input a table where each row contains a packet id, packet length, packet queue time.

Knowing this, it is possible to cluster the packets length in small interval and count the average waiting time of each packet of that cluster.

The result of the Matlab computation is then plotted on a histogram and the result is showed in picture 1.

# 3.3 – Average response time

Average response time is computed from when the packet is created, to when the packet is destroyed. In the simulation it is computed by Omnet, and it is:

# 3.4 – Utilization factor of the server

Utilization factor of the server is the percentage of time the server is working over all the simulation time.

This also is computed by Omnet, and it is:

# 3.5 – Queue length over time

Queue length over time data is collected by Omnet and are plotted in figure 2.

# 4 – Compare experimental and theoretical values

Once experimental values are collected after running a simulation, they are compared to theoretical values calculated by solving equations of an M/G/1 system.

# 4.1 – Average conditional queue time

COME SI CALCOLA?

# 4.2 – Average response time

In an M/G/1 system, average response time is:

where is the expected value of a uniform distribution, and is the NNNNNNNNN.

Using numerical values in the formula give us:

which is a bit greater than the experimental result.

# 4.3 – Utilization factor of the server

As above, the formula for server utilization factor in an M/G/1 system is:

which is perfectly fine with the experimental result.