# Project B

DT057A: Performance Analysis and Simulation of Communication Systems

# Part 1

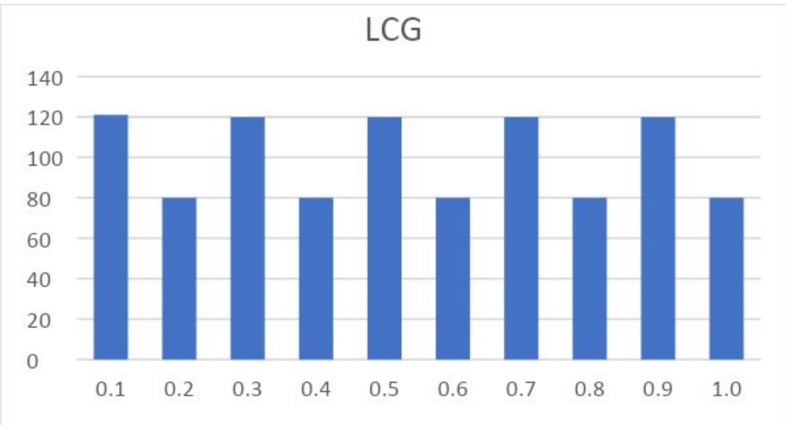
1. **Create a function that implements a linear congruential generator (LCG), accepting as input the parameters: seed, m, a and c.**

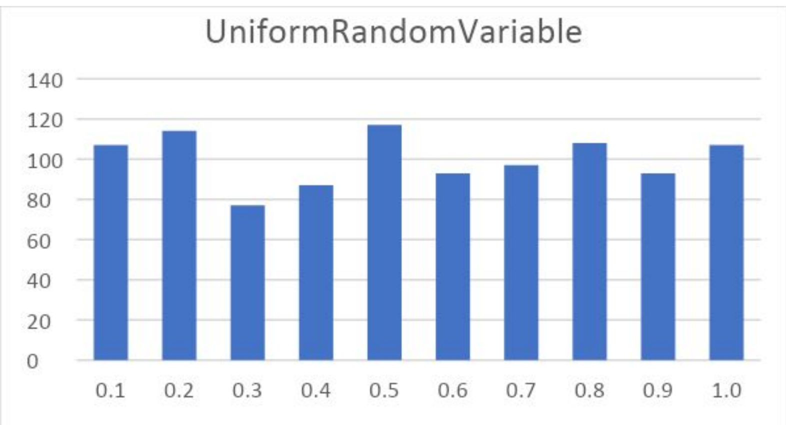
   **Hint: It is better if you do not attempt to modify the rng module of ns-3 instead create a function in your simulation file (e.g. mysimulation.cc) and call the function from the main.**

2. **Generate 1000 values uniformly distributed in the range [0,1] using your PRNG. For this case use m=500, a=3 c=0 and seed =1;**

3. **Compare the distribution of your values with the distribution of values generated using the UniformRandomVariable()of ns-3.**

   The sequence of numbers is repeated after about 100 random values in the LCG distribution. Whereas no repeated sequences are found among the output of UniformRandomVariable.




4. **Comment on the difference in the results and propose values of m, a and c which gives you better results.**

The recurrence relation works well with a high modulo value. The more values wanted as output, the higher the value of the modulo value. With a value of m=50 000 no repeating sequences within the 1000 outcome values were found. Prime numbers may be preferable.

5. **What PRNG does ns-3 use? What method does ns-3 use to generate a normal random variable?**
   Ns3 uses ns3::RandomVariableStream as PRNG. To generate random variables smart pointers are used together with using the templated CreateObject<> method.

   *Ptr<UniformRandomVariable> x = CreateObject<UniformRandomVariable> ();*

6. **Using the time system command of linux compare the execution time for the generation of the uniform distribution using your function and ns-3 function.**

   LCG: real time = 6.47

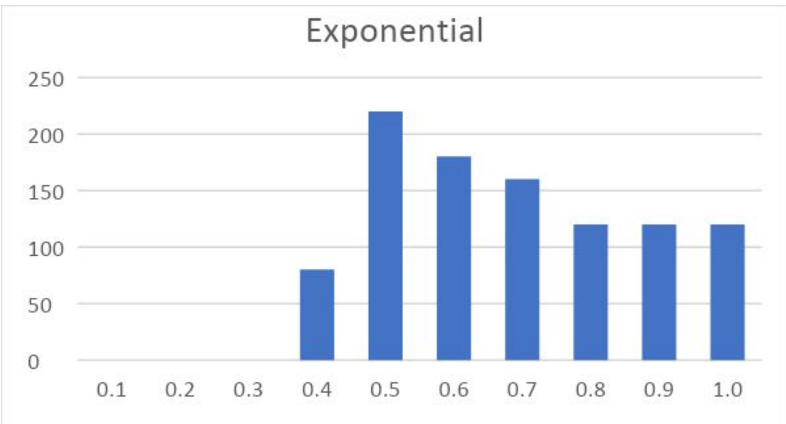
   URV: real time = 2.25

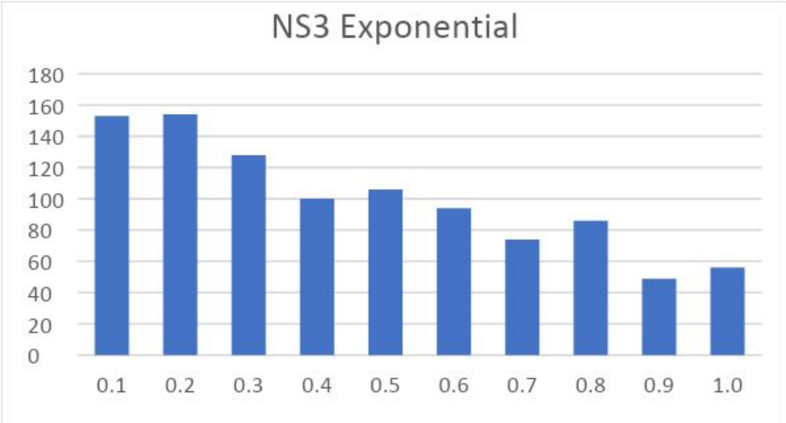
   The generation of the LCG took about three times as long as UV, probably because of URV being optimized.

7. **Write a second function that generate an exponential distribution with mean $\beta > 0$ from a uniform distribution generated using the LCG; Choose one of the methods for generating RV covered in the course and motivate your choice with respect to the specific task.**

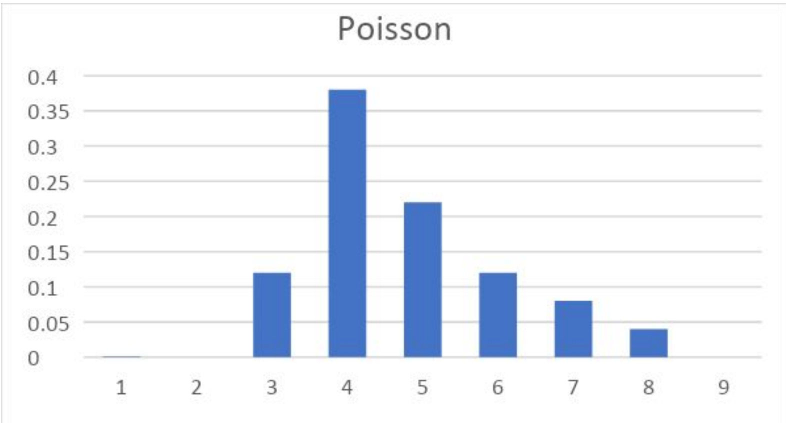
8. **Compare your exponential distribution with ns-3 ExponentialRandomVariable() and the theoretical expression of the probability density function.**

$$p\left(x|\lambda\right) = \lambda e^{-\lambda x}, x > 0$$

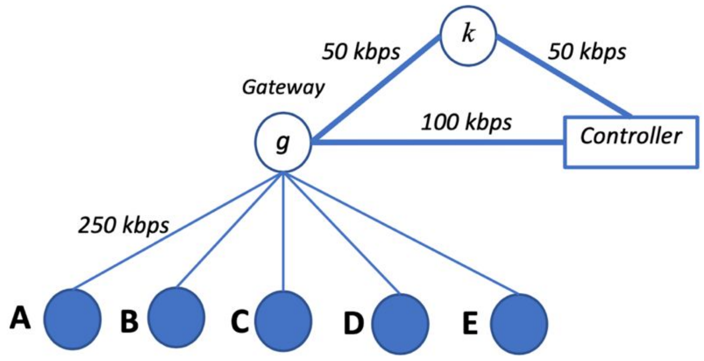





9. **Using your exponential random variable create a function that generate Poisson distributed random numbers with mean $\lambda > 0$.**



# Part 2




**What is the average number of sensor messages in the system that are traveling to the controller?**

$$1\ packet\ =\ 50\ Byte\ =\ 50 * 8\ bit\ =\ 400\ bit$$

## Service time:

$$\frac{1}{\mu_{A,B,C,D\rightarrow gateway}} = 400bit/250\ kbps\ =\ \frac{400}{250\ 000}\ s\ =\ 0.0016\ s\ =\ 16ms$$
$$\frac{1}{\mu_{gateway\rightarrow controller}}\ =\ 400\ bit/100\ kbps\ =\ \frac{400}{100\ 000}\ s =\ 0.004\ s\ =\ 4ms$$
$$\frac{1}{\mu_{gateway\rightarrow k}}\ =\ 400bit/50\ kbps\ =\ \frac{400}{50\ 000}\ s\ =\ 0.008\ s\ =\ 8ms$$
$$\frac{1}{\mu_{k\rightarrow controller}}\ = 400bit/50\ kbps\ =\ \frac{400}{50\ 000}\ s\ =\ 0.008\ s = 8ms$$

## Service rate:

$$\mu = \frac{1}{Service\ time}$$

$$\mu_{A,B,C,D\rightarrow gateway}\ =\ \frac{1}{0.0016}\ =\ 625\ packets/s$$
$$\mu_{gateway\rightarrow controller}\ =\ \frac{1}{0.004}\ =\ 250\ packets/s$$
$$\mu_{gateway\rightarrow k}\ =\ \frac{1}{0.008}\ =\ 125\ packets/s$$
$$\mu_{k\rightarrow controller}\ =\ \frac{1}{0.008}\ =\ 125\ packets/s$$

## Arrival rate:

New packet arrives every 20 ms. (Same for all links)

$$1/20\ =\ 0.05\ packets/ms\ =\ 50\ packets/s$$

$$\lambda = 50\ packets/s$$

## Average no. of packets in each link:

$$N_{ij}\ =\ \frac{\lambda_{ij}}{\mu_{ij}-\lambda_{ij}}$$

$$N_{A,B,C,D\rightarrow gateway}\ =\ \frac{50}{625-50}\ = 0.0869\ packets/s$$

$$N_{gateway \to controller} = \frac{50}{250-50} = 0.25\ packets/s$$

$$N_{gateway \to k} = \frac{50}{125-50} = 0.6667\ packets/s$$

$$N_{k \to controller} = \frac{50}{125-50} = 0.6667\ packets/s$$

Average no. of packets in system (traveling to controller):

$$\sum N_{ij} = 0.08696 * 4 + 0.25 + 0.667 * 2 = 1.9312\ packets/s$$

**What is the average number of queued messages in the gateway waiting for transmission to the controller?**

$$\mu_{A,B,C,D \to gateway} = 625\ packets/s$$

$$\lambda_{A,B,C,D \to gateway} = 50\ packets/s$$

$$N_{A,B,C,D \to gateway} = \frac{50}{625-50} = 0.0869\ packets/s$$

$$\sum N_{ij} = 0.08696 * 4 = 0.3478\ packets/s$$

Probability of gateway forwarding packet to the controller:

$$p = 0.9$$

Average number of queued messages transmitted directly to controller:

$$0.3478 * 0.9 = 0.313\ packets/s$$

**If the buffer size at gateway for the communication link between gateway and controller is of size K=10, what is the percentage of packet that are lost due to a full buffer?**

Service time (Gateway  Controller): $\frac{1}{\mu} = 4\ ms = 0.004\ s/packet$

Service rate: $\mu_{gateway \to controller} = \frac{1}{0.004} = 250\ packets/s$

Arrival rate: $\lambda = 50\ packets/s$

Blocking probability function:

$$P_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}$$

Utilization factor:

$$\rho = \frac{\lambda}{\mu} = \frac{50}{250} = 0.2$$

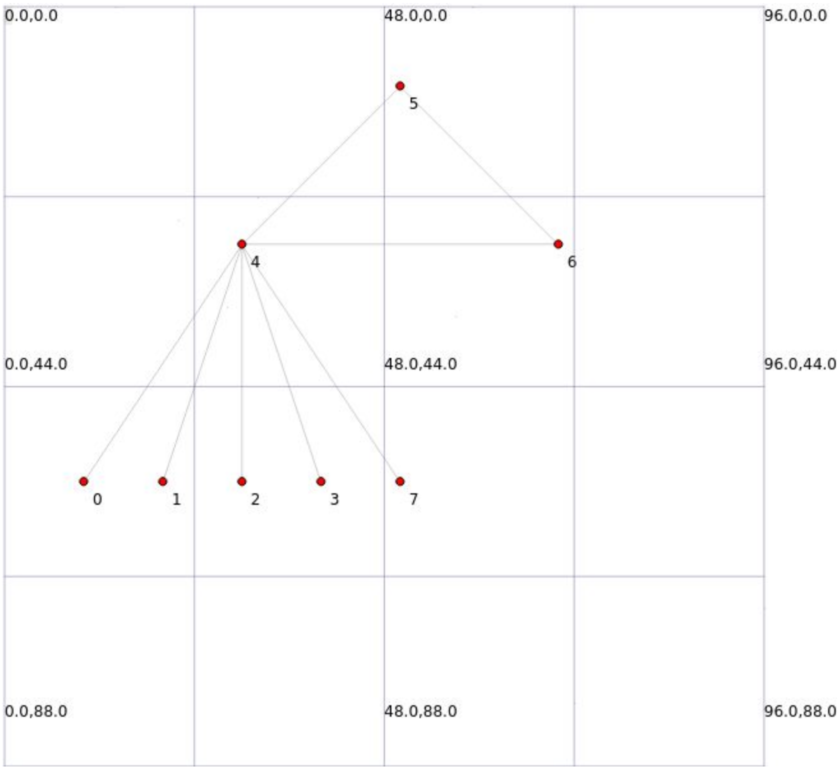
Buffer size: K = 10

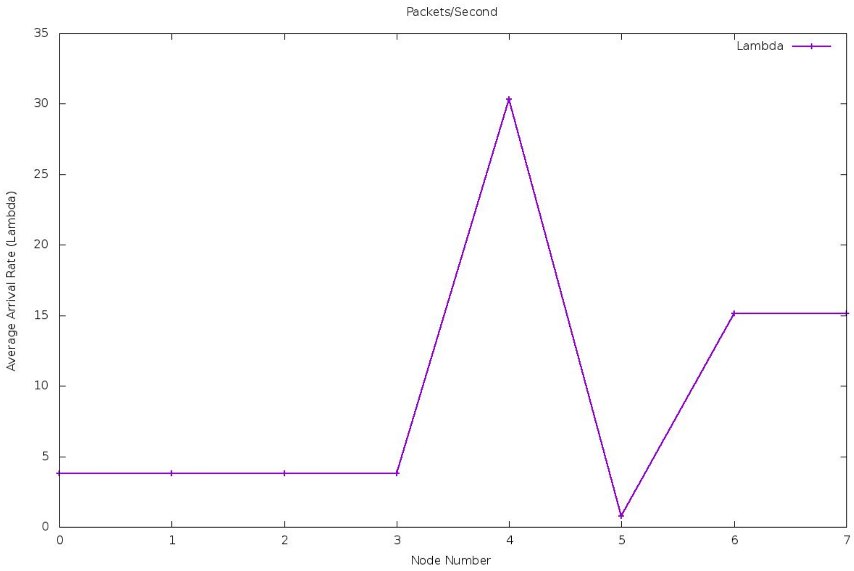
$$P_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} = \frac{(1-0.2)*0.2^{10}}{1-0.2^{11}} = 8.19 * 10^{-8}$$

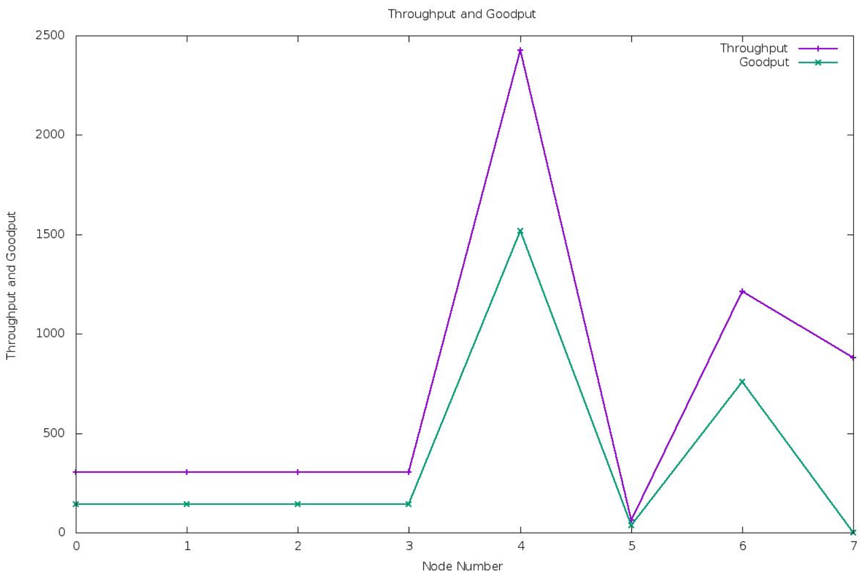
## Part 3

*Implement the scenario in ns-3. Use the full duplex P2P communication for all the links. For the generation of the traffic you can use the UDP Client and Server Applications.*

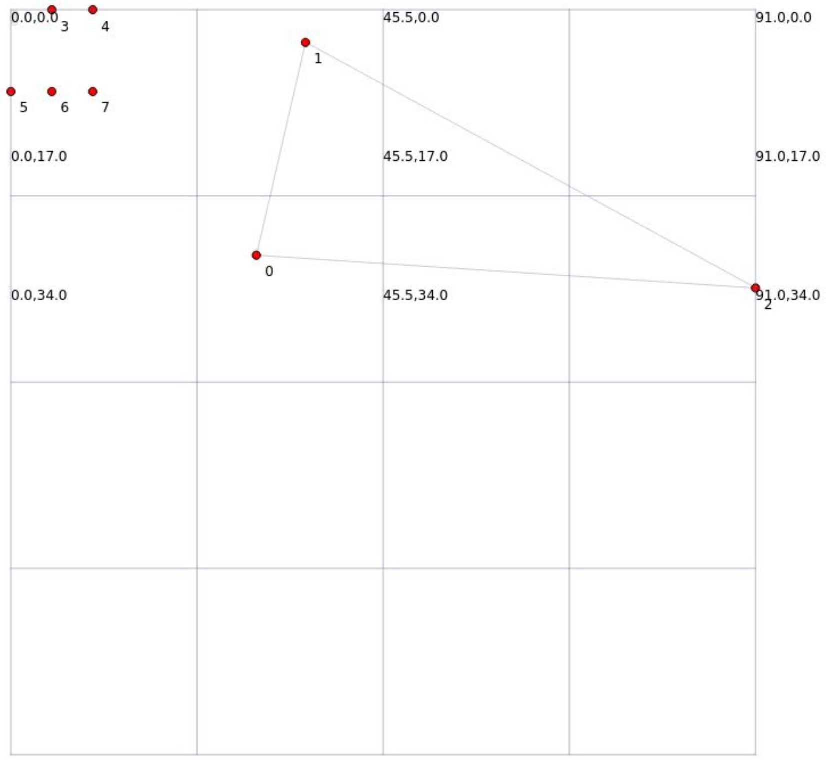
P2P Communication




Packets/Second
Lambda
Average Arrival Rate (Lambda)
Node Number
0
5
10
15
20
25
30
35
0
1
2
3
4
5
6
7


Throughput and Goodput
Throughput
Goodput
Throughput and Goodput
Node Number
0
500
1000
1500
2000
2500
0
1
2
3
4
5
6
7

# Wireless Communication




# Simulation Summary

```
Lines on file:           243
Total enqueued packets:     81
Total sent packets:     81
Total received packets:     81
Total dropped packets:     0
Total simulation time:     2.56891 seconds
Time of analysis:     0s
```

# Nodes

```
Node:          0
Sent packets:      40
Received packets:      40
Dropped packets:      0
Data sent:          3.125 KB
Data received:      3.125 KB
Data dropped:      0.0 B
Throughput:         1245.664503622159 B
Goodput:            778.5403147638493 B
```

```
Lambda:      15.570806295276986
EN:          0.3402454737612454
EW:          0.021851500000000044
Little's result:
  -> EN:     0.3402454737612454
  -> EW*lambda:    0.34024547376124575
Average length of:
  -> Sent packets:     80.0 B
  -> Received packets:     80.0 B

Node:         1
Sent packets:     1
Received packets:     1
Dropped packets:     0
Data sent:         80.0 B
Data received:     80.0 B
Data dropped:     0.0 B
Throughput:        31.141612590553972 B
Goodput:           19.463507869096233 B
Lambda:      0.38927015738192466
EN:          0.0
EW:          0.0
Little's result:
  -> EN:     0.0
  -> EW*lambda:    0.0
Average length of:
  -> Sent packets:     80.0 B
  -> Received packets:     80.0 B

Node:         2
Sent packets:     40
Received packets:     40
Dropped packets:     0
Data sent:         3.125 KB
Data received:     3.125 KB
Data dropped:     0.0 B
Throughput:        1245.664503622159 B
Goodput:           778.5403147638493 B
Lambda:      15.570806295276986
EN:          0.07344749329482161
EW:          0.004716999999999949
Little's result:
  -> EN:     0.07344749329482161
  -> EW*lambda:    0.07344749329482074
Average length of:
  -> Sent packets:     80.0 B
```

```
  -> Received packets:     80.0 B
```

## Throughput/Goodput

```
0    1245.664503622159    778.5403147638493
1    31.141612590553972    19.463507869096233
2    1245.664503622159    778.5403147638493
```

## Little's Result

```
0    15.570806295276986    0.021851500000000044    0.3402454737612454
0.34024547376124575
1    0.38927015738192466    0.0    0.0    0.0
2    15.570806295276986    0.004716999999999949    0.07344749329482161
0.07344749329482074
```

## Streams

```
UDP ALL:
------ Stream 0 ------
Ips:         10.1.3.2 --> 10.1.3.1
Ports:       49153 --> 10
Number of:
  -> Sent packets:     40
  -> Received packets:     40
  -> Dropped packets:     0
  -> Drop sequences:     0
Average drop:     0.0
Drop Variance:     0.0

------ Stream 1 ------
Ips:         10.1.2.1 --> 10.1.2.2
Ports:       49153 --> 9
Number of:
  -> Sent packets:     1
  -> Received packets:     1
  -> Dropped packets:     0
  -> Drop sequences:     0
Average drop:     0.0
Drop Variance:     0.0

------ Stream 2 ------
```

```
Ips:          10.1.3.1 --> 10.1.3.2
Ports:        49154 --> 9
Number of:
  -> Sent packets:     39
  -> Received packets:     39
  -> Dropped packets:     0
  -> Drop sequences:     0
Average drop:     0.0
Drop Variance:     0.0

------ Stream 3 ------
Ips:          10.1.1.1 --> 10.1.1.2
Ports:        49153 --> 9
Number of:
  -> Sent packets:     1
  -> Received packets:     1
  -> Dropped packets:     0
  -> Drop sequences:     0
Average drop:     0.0
Drop Variance:     0.0
```