Performance Analysis and Simulation of Communication Systems: Project

April 24, 2020

The project consists of a short technical report, and an oral presentation. The report must be written in English.

The students are encouraged to work in groups of three people. Each group must communicate the project that they have selected by writing an email to the instructor. The email should contain:

1. the name of the students in the group;
2. the project that they have chosen (A, B or C).

Project Deadline:

There is no deadline for submitting the project.
Submitting the project report is required to pass the project (and therefore to ultimately pass the course).

Project Grade:

The project grade consists in a Pass or Fail.

Project report

The extent of the report should correspond to at least twenty (8) tightly written or newly written pages (without the cover page and the pages with the content, the abstract, the list of figures and the references), plus illustrations, cover page and list of contents. It is recommended that the report includes many illustrations (for example screen shots), tables (for example comparison tables and measurement results) and/or plots (with measurement results). One of the students in each group should upload an electronic version of the final version of the report on the moodle page of the course before the presentation. The content of the project report should be similar to the content of the presentation. The students should motivate their choices regarding the method of implementation, the simulation parameters and the models used (traffic, channel, mobility, etc.). The report should contain an accurate description of the methods and experiments used to obtain the results. The report should not contain copied text (unless it is clearly marked that it is a citation, stating the source where it is taken from). Instead, you should use your own formulation and conclusions. Copying text means that your work is not genuine, but a replica and this can have consequences.

The report should describe the simulated scenario (i.e. a list of models and of their parameters). The report should contain the results of the simulation in form of graphs (line graph, bar graph, etc.). The results should be inclusive of summary statistics (e.g. mean, variance) and confidence interval. The students should detail which methods they have used to record, calculate, and present data and statistics. Other information to include are the number of simulation run, duration of each simulation, Warm-Up time (if needed).

The report should end with description of sources you have used.

Project Presentation

The students should prepare a short 10-minute presentation. The presentation will be given remotely, using zoom (https://miun-se.zoom.us/my/lucbel) or skype at the discretion of the students. During their presentation the students should describe the simulated scenario (i.e. a list of models and of their parameters). Each group, previous agreement with the examiner, can decide the date and time of the presentation. All students in a group have to attend the presentation, only in case that was not possible a student may ask to present separately.

For questions, you can send an email at **luca.beltramelli@miun.se**.

Project A

Part I: PRNG and Random Variables

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*=100, *a*=13 c=1 and seed =1;
3. Compare the distribution of your values with the distribution of values generated using the `UniformRandomVariable()` of ns-3.
4. Comment on the difference in the results and propose values of *m*, *a* and *c* which gives you better results.
5. What PRNG does ns-3 use? What method does ns-3 use to generate a normal random variable?
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.

7. Write a second function that generates 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.

Part II: Mathematical Modelling of a system of Queues

The system we model represents a typical network infrastructure interconnecting a subnet with the Internet. In the scenario, shown in figure 1, there are four sources of traffic, named *A*, *B*, *C* and *D*.

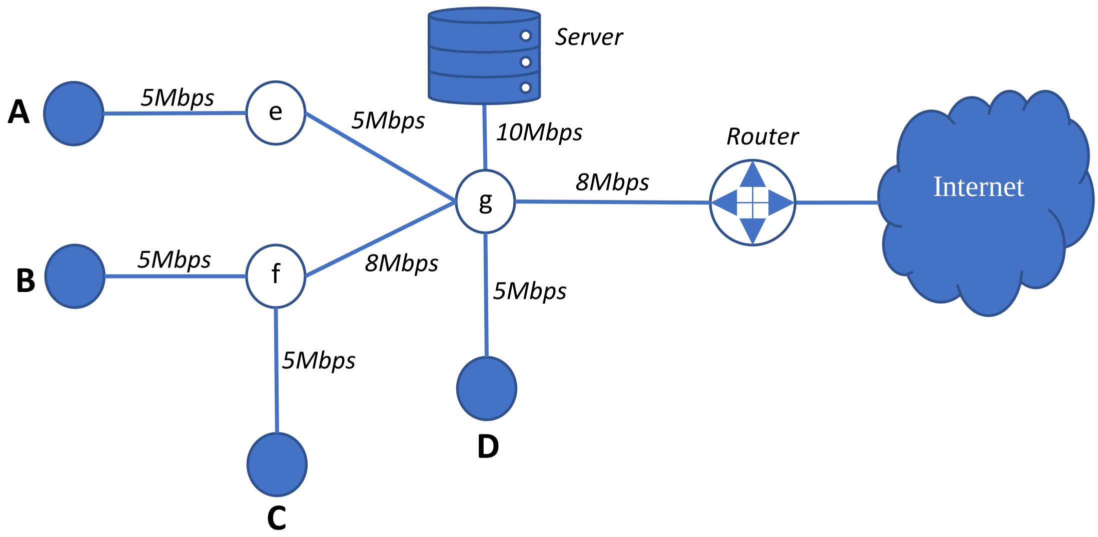

Figure 1: Scenario for Project A

Assume that each of the four devices (*A*, *B*, *C* and *D*) generates request packets with a Poisson distribution (i.e. an exponential distributed time between packets). The size of each packet is also exponentially distributed. See **Table I** for the exact values.

Table 1 Traffic parameter for Project A

Source	Mean packet interarrival time	Mean packet size
A	2 ms	100 Bytes
B	2 ms	100 Bytes
C	0.5 ms	100 Bytes
D	1 ms	100 Bytes

The server with probability *p=0.7* will immediately reply to a received packet with a response packet addressed to the source of the request packet.

With probability *1-p* the server will forward the request through the router to the internet.

All the communication links are full duplex and Point-to-Point, you can assume the signal propagation delay to be negligible.

Before transmission each packet is buffered in a queue, each device has a separate queue for each P2P link. For example, device *f* has three queues one for the packets going to *B*, one for the packet going to *C* and one for the packet going to *g*.

Using queuing theory and the network of queues theory answer the following questions:

1. What is the average number of request packets in the system that are traveling to the server?
2. What is the average number of request packets in each link that are traveling to the server?

Part III: Simulations and Results Comparison

Implement the scenario in ns-3. Use the P2P communication for all the links. For the generation of the traffic you can use the UDP Client and Server Application. The Server application will need to be modified so that it reply only to a fraction *p=0.7* of the messages it receives while the rest is forwarded to the router. You can assume that the router will simply drop all the received messages without taking any action.

- Run the simulation and measure:
  - The average number of request packets queued at the link between *g* and the *Router* (*g*→ *Server*)
  - The average queuing delay and the total average delay for the request packets traversing the link f →*g*
- Compare the results you get from the simulation with the one you obtained in Part II.
- Measure the delay between *A* and the *Server* and between the *Server* and *A*.
- Replace each P2P link with a bus using CSMA/CD do not change the datarate or delay.
- Run the simulation and compare the results with the one obtained with the P2P. Try to explain the difference you see.
- Run the simulation but this time use the custom PRNG that you implemented in Part I to generate the exponential packet size and the exponential time between packets. Which differences do you observe?

Project B

Part I: PRNG and Random Variables

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.
4. Comment on the difference in the results and propose values of *m*, *a* and *c* which gives you better results.
5. What PRNG does ns-3 use? What method does ns-3 use to generate a normal random variable?
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.
9. 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.
10. Compare your exponential distribution with ns-3 `ExponentialRandomVariable()` and the theoretical expression of the probability density function.
11. Using your exponential random variable create a function that generate Poisson distributed random numbers with mean $\lambda > 0$.

Part II: Mathematical Modelling of a system of Queues

Consider the system showed in figure 2. It consist of a WSAN (Wireless Sensor and Actuator Network), in which nodes *A*, *B*, *C*, *D* are the sensors and node *E* is the actuator. All nodes communicate with a gateway *g* forming a star topology. The gateway forward the messages from the sensors to the controller and the messages from the controller to the actuator (*E*). With probability *p=0.9*, the gateway forward the sensor messages directly to the controller. With probability *1-p,* the gateway forward the sensor messages to the controller through node *k.* The controller instead transmits all its messages for the actuator (*E*) directly to the gateway.

The links between the gateway and the controller, controller and node *k,* and, node *k* and gateway, are wired full duplex P2P. The communication between the nodes and the gateway is wireless. All messages are generated with an exponential size. The messages are generated according to a Poisson random variable, that is the time between the generation of two messages is exponentially distributed. The mean of the message generation interval and message size are available in Table II.

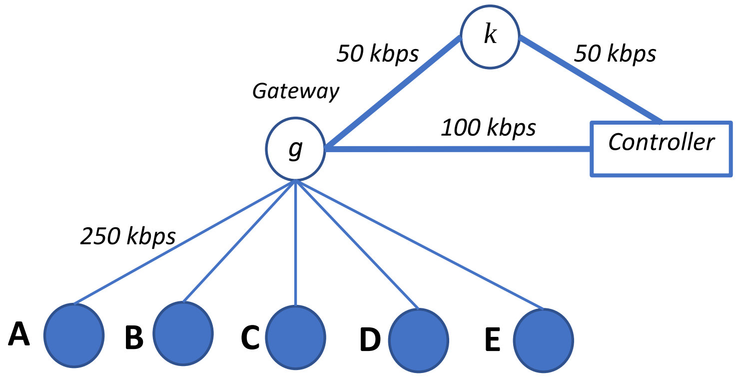

Figure 2: Scenario for Project B

Table 2 Traffic parameter for Project B

Source	Destination	Mean time between packets	Mean packet size
Sensors (*A, B, C, D*)	*Controller*	20 ms	50 Bytes
Controller	Actuator (*E)*	20 ms	50 Bytes

Assume at first that all the wireless links are full duplex P2P with a datarate of 250kbps each. Before transmission each packet is buffered in a queue, each device has a separate queue for each P2P link. For example, device *g* has three queues one for the packets going to *E*, one for the packet going to *k* and one for the packet going to the *Controller*.

Using queueing theory and the network of queues theory answer the following questions:

- What is the average number of sensor messages in the system that are traveling to the controller?
- What is the average number of queued messages in the gateway waiting for transmission to the controller?
- 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?

Part III: Simulations and Results Comparison

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.

- Run the simulation and compare the results you get from the simulation with the one you derived in part II.

- Replace all the P2P links between nodes and gateway with wireless channel at 2.4 GHz with datarate of 250 kbps **based on the standard IEEE 802.15.4 (ns-3 LR-WPAN models)**.
- Use a channel and mobility model that you want.
- Measure the end-to-end delay between sensor *A* and the *Controller* and the delay between *Controller* and actuator (*E*).
- **Run the simulation and compare the new results with the one obtained by using the P2P model. Motivate the difference in the results.**
- **Run the simulation but this time use the custom PRNG that you implemented in Part I to generate the exponential packet size and the exponential time between packets. Which differences do you observe?**

Project C

Part I: PRNG and Random Variables

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=4 and seed =1;
3. Compare the distribution of your values with the distribution of values generated using the `UniformRandomVariable()` of ns-3.
4. Comment on the difference in the results and propose values of *m*, *a* and *c* which gives you better results.

5. What PRNG does ns-3 use? What method does ns-3 use to generate an exponential random variable?
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.
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.

Part II: Mathematical Modelling of pure ALOHA

Consider the system showed in figure 3, consisting of N devices (n_1,n_2,n_3, n_4,…, n_N) communicating with node *A* and *B* through a gateway.
The devices generate packets for *A* and *B*. All devices generate messages at the same rate; the aggregate rate at which the N devices combinaed generate messages is of 100 message/s (50 message/s to *A* and 50 message/s to *B*).
The time between the generations of messages is exponentially distributed (i.e. Poisson traffic). All devices(n_1,n_2,n_3, n_4,…, n_N) access the wireless channel using pure ALOHA. The gateway communicates to A and B through two P2P links with datarate of 400 kbps each. All packets have the same fixed size of 100 Bytes. All messages are sent without acknowledgment, that is, messages that are lost or not correctly received are not retransmitted.

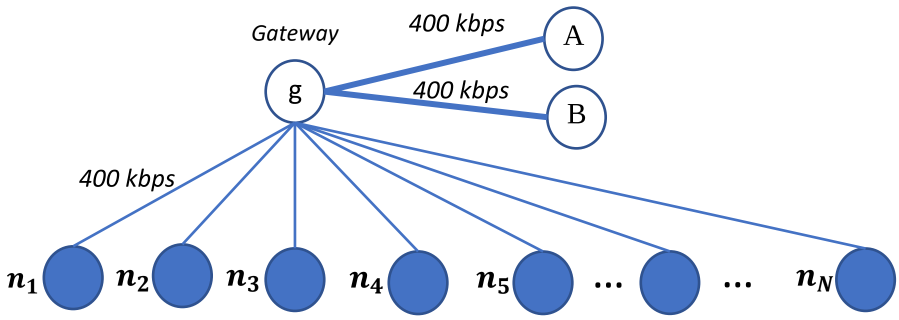

Figure 3: Scenario for Project C

- What is the throughput of the wireless network for $N \to \infty$? What is the offered traffic G?
- For $N \to \infty$, what would be the throughput of the wireless network if we were to use Slotted ALOHA?
- Assuming to replace the wireless links with dedicated P2P links with a datarate of 400kbps each, what is the average number of messages stored in the gateway waiting to be transmitted to *A* and *B*? (answer using queueing theory)

Part III: Simulation and Result Comparison

Implement the scenario in ns-3.

- Implement the scenario in ns-3, for reference on how to implement the ALOHA channel access check the examples you find in the folder `src/spectrum/examples/`.
- Use a channel and mobility model that you want but be sure that the only cause of packet loss are the collision.
- Run the simulation and measure the throughput and packet loss of the system.
- Compare the theoretical results obtained in Part 2 with the results you obtain by simulating the scenario with 10, 20, 30, …, 90, 100 devices (Remember that even when increasing the number of devices the aggregate rate of generation of messages should always be of *100 packet/s*). How does the difference between theoretical and simulation result changes by varying the number of nodes? Why?
- Replace the channel and mobility models so that even when packets do not collide they could be lost. Run the simulation and compare the results.
- Run the simulation but this time use the custom PRNG that you implemented in Part I to generate the exponential time between packets. Which differences do you observe?