13 Simulation exercise 2017

Smart meters will be installed in all households in Norway by Jan 1, 2019. The purpose is to collect information about power consumption[1]. The meter data will be collected by the power grid operators and forwarded to ELHUB, a centralised database for Norway.

For this we need a communication system. In Figure 1, a simplified sketch of a system is given, where the smart meters are connected to a mobile network operator, and the smart meter data is forwarded to via the mobile network to the centralised data centre (ELHUB)[2].

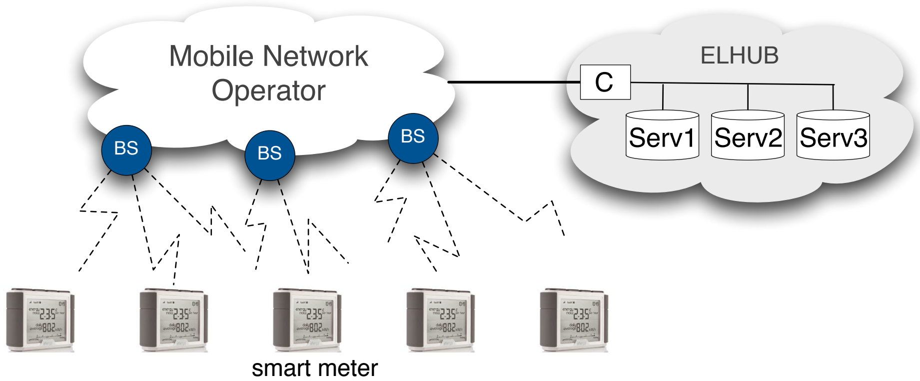

Figure 1: Communication system for transferring meter readings from smart meters to the ELHUB

Each smart meter sends its meter readings as data packets periodically to the ELHUB. A packet is sent from the smart meter to the base station (BS) which is associated with the mobile unit in the smart meter. Then the packet is forwarded through the transport network to the ELHUB controller (C), which is responsible for routing an incoming data packet to one of the two active servers. The ELHUB data center consist of three servers, two active and one backup. If an active server fails, the passive server will replace the failed server. The data packet is processed by the server before it is stored.

Assumption A: The data packets from a smart meter (SM) are stemming from a Poisson process with intensity λ per meter.

Assumption B: No data packets are lost in the BSs and they never fail

Assumption C: The data packets from the smart meters are lost on the wireless link between the SM and BS (i.e., not received by the base stations (BS)) with the probability $1 - p_r$.

Assumption D: The radio link delay between the smart meter and the associated BS is constant T_w=$(1/\mu_w)$.

Assumption E: The delay through the transport network between the BS and the controller (C), is T_n ∼n.e.d.(μ_n).

Assumption F: Processing of a data packets in C is T_c ∼n.e.d(μ_c)

Assumption G: Processing of a data packets in a server is T_s ∼n.e.d(μ_s)

Assumption H: The servers fail independently of each other with a constant failure intensity λ_{fs}.

Assumption I: The time to restart a server is T_{rs} ∼ n.e.d(μ_{rs})

Assumption J: The controller (C) fails with a constant failure intensity λ_{fc}.

Assumption K: The time to restart the controller is T_{rc} ∼ n.e.d(μ_{rc})

Assumption L: The time to switchover from passive to active server is T_{sw} ∼ n.e.d(μ_{sw})

Assumption M: The data centre is working when at least one server and the controller (C) are working.

Assumption N: No data packets are lost in the transport network and it never fails

[1] Plus a few other features not covered in this assignment, such as status information about power grid, send price signals, etc.
[2] This is not exactly how it is implemented but it's simplified for convenience.

PART I: System performance [50 Points]

In the first part of this assignment the objective is to study the *data packet delay*, T under the assumption that, BS, transport network, C, and two active servers are always working.

Make a model with the objective to estimate the average packet delay, T, and the probability that $P(T < 200)$ which is the requirement given for the centralised ELHUB.

1. [5 Points] What is the *system state* and corresponding *events* in your model?
2. [15 Points] Draw the activity diagram for the system.
   Include all statements needed for collecting data for the statistics (TALLIES etc.).

Implement the simulation model with Simula and the DEMOS class.
Have a look at Appendix A for some SIMULA hints.

3. [10 Points] Run your simulator for at least 100000 date packets, using the following parameters:

$T_w = 1/\mu_w$	110 [ms]
$T_n = 1/\mu_n$	20 [ms]
$T_c = 1/\mu_c$	0.1 [ms]
$T_s = 1/\mu_s$	2 [ms]
p_r	0.95
λ	To Be Defined by you
BS	50

   Observe!! The parameter λ is the intensity of packets from one single smart meter. This should be set by you such that the system is not overloaded (a unstable system is created), but at the same time demonstrates that the number of smart meters and number of servers (significantly) affects the performance.

Assume that the BSs and smart meters are geographically uniformly distributed such that the number of smart meters per BS is the same for all BSs. Then extend the number of smart meters $N_{sm} = 1000, 10000, 100000$.

4. [10 Points] Run a series of simulation experiments for three values of N_{sm}. Change the number of available servers from $N_s = 2$ to $N_s = 4$ and repeat the experiments. Plot the estimated packet delays for each value N_{sm} and for both values of N_s in the same plot. Can you (briefly) explain the results you observe?

Implement a logging function in your simulator. Observe each packet sent from the SM to ELHUB, and log the packet ID and the end-to-end delay of the packets (i.e., the time from the packet was sent from the SM until it was stored in the ELHUB). Take into account that a packed might be lost along the way.

5. [10 Points] From your log: estimate the probability of a packet loss, and plot the Cumulative Distribution Function for the end-to-end packet delay.

PART II: ELHUB availability [50 Points]

In this part we assume that the ELHUB consists of two active and one passive server, which all might fail with failure intensity λ_{fs}. When an active server fails, the controller initiates a switchover (T_{sw}) where the failed active server is replaced with the passive. The controller might also fail with failure intensity λ_{fc}. The parameters are listed in the list of assumptions, and the numerical values in the table below. The objective is to assess the probability that the ELHUB is unavailable.

1. [5 Points] What is now the *system state* and corresponding *events* in your model? When is the ELHUB unavailable?
2. [20 Points]. Draw the activity diagram for the system. Include all statements needed for collecting data for the statistics (TALLIES etc.). Implement the simulation model by use of Simula and the DEMOS class.
3. [15 Points] Run your simulator with parameters for the dependability of the contollers and servers in the ELHUB (see Table).

parameter	value [hours^{-1}]
λ_{fs}	0.01 (or 0.1)
μ_{rs}	1
λ_{fc}	0.01
μ_{rc}	2
μ_{sw}	5 (or 0.3)

Alternative parameters $\lambda_{fs} = 0.1$ and $\mu_{sw} = 0.3$ (with the other parameters in the table unchanged) will give more interesting results where the unavailability of the servers in the cases with and without switchover time is significant, and the total unavailability is not dominated by the unavailability of the controller.

An analytical expression for the unavailability of the server and the controller in the ELHUB can be determined using Markov models (see textbook, Chapter 5 and 7). An approximate expression for the unavailability of the servers (which are available if at least one server is active) is:

$$U_s \approx \frac{\lambda_{\mathrm{fs}}^2 \left(\lambda_{\mathrm{fs}} \left(4\mu_{\mathrm{rs}}^2 + 9\mu_{\mathrm{rs}}\mu_{\mathrm{sw}} + 3\mu_{\mathrm{sw}}^2\right) + 6\mu_{\mathrm{rs}}^3\right)}{3\mu_{\mathrm{rs}}^2 \left(\mu_{\mathrm{rs}} + \mu_{\mathrm{sw}}\right) \left(\lambda_{\mathrm{fs}} \left(8\mu_{\mathrm{rs}} + 3\mu_{\mathrm{sw}}\right) + \mu_{\mathrm{rs}} \left(2\mu_{\mathrm{rs}} + \mu_{\mathrm{sw}}\right)\right)} \tag{1}$$

If the switchover time is $1/\mu_{sw} = 0$ then the unavailability of the servers is

$$U_s = \frac{\lambda_{\mathrm{fs}}^3}{\left(\lambda_{\mathrm{fs}} + \mu_{\mathrm{rs}}\right)^3} \tag{2}$$

The unavailability of the controller is:

$$U_c = \frac{\lambda_{\mathrm{fc}}}{\lambda_{\mathrm{fc}} + \mu_{\mathrm{rc}}} \tag{3}$$

If we assume that the server and controller fail and are repaired independent of each other, then the total unavailability of the ELHUB is:

$$U = 1 - (1 - U_s)(1 - U_c) = U_s + U_c - U_s U_c \tag{4}$$

4. [10 Points] Use the results from you simulations and cross-validate against the (approximative) analytic unavailability given by Equation (4).

General comments:

Note that this is not a pure programming exercise. Simulation is only a tool to achieve the objectives. You are also expected to carefully plan the experiment (definition of model, activity diagram etc. as noted in the tasks) and explain what the results mean for the objectives. If something is not defined detailed enough, you are expected to make educated assumptions. Write down your assumptions in the report and justify them.

If you are in doubt, you may ask the course staff. They are happy to clarify ambiguities, however, they will not help you solve the assignment and you are expected to be well prepared, i.e. have studied and understood the simulation exercises from the preparation week.

Appendix

A. SIMULA hints

A.1. How to run replications:

preamble

```
INTEGER  numreplica;
```

code

```
...
numreplica:=2;
! replications ;
WHILE replication LE numreplica DO
   BEGIN
      ...
      ! <all entities are scheduled here>;
      ...
      ! limit the output by not printing the end report for each replication
      (only the accumulated one) if numreplica>2;
   IF numreplica > 2 THEN noreport;
   replicate;
END;
! for less than 3 replications do not print a replication report;
IF numreplica < 3 THEN norepli;
```

A.2. How to pass arguments to a class:

This example is from Exercise 6, see the solution of it for more details.

code

```
...
 Entity class Router(inputBuffer_);
         ref(WaitQ) inputBuffer_;
         begin
          ...
         end;
```

A.3. How to terminate simulation:

In the exercises 5-7 we used a *time controlled termination* of the simulation, i.e. the simulation ran for a certain SIMULATION_TIME. This was implemented by writing `hold(SIMULATION_TIME)` at the end of the code. If you want an *event controlled termination*, i.e. terminate the simulation after *NSim* events, add the following code. The example is for Model B from the lecture.

code

```
begin
  external class demos="./demos.atr";
  demos BEGIN
        REF(BIN)  finished_packets;
        INTEGER NSim; ! number of events to be simulated
        ...
        Entity class Router;
                begin
                  ...
                  ! end of packet activities;
                  finished_packets.give(1);
        end;
        finished_packets :- NEW BIN("finished packets", 0);
```

```
        ...
        finished_packets.take(NSim);
  end;
end;
```

A.4. Common Mistakes:

In SIMULA, a comment starts with a "!" and ends with the next ";" (not a line break!), i.e.:

code

```
this is code ! this is
all comment
until
here;
```

A.5. Runtime Errors:

If you get a *segmentation fault* or a *System error: Bus error*, try compiling your code with more memory (number is in MB):

terminal

```
cim -m10 my_code.sim && ./my_code
```

A.6. Print to screen:

terminal

```
outtext("Tekst");
outint(tall,10);                   ! print integer with 10 positions;
outfix(tall,4,10);          ! print real with 4 decimales and 10 positions;
outreal(tall,4,20);          ! print real scientific format;
outimage;                          ! end of line;
```