Simulation of Data Networks S-38.3148

CNCL Exercise - Fall 2007

Contents

Chapter 1

Introduction

The purpose of this exercise is to make you familiar with CNCL simulation library. You will learn how to create new simulator blocks with C++ by using the objects and functions provided by CNCL and how to collect statistics from the simulation.

The exercise deals with modelling mobility in adhoc network simulations. Random Waypoint (RWP) mobility model is one of the most often used mobility models. In this exercise connectivity of nodes moving according to this model is studied.

Chapter 2

Background for the exercise

2.1 Random Waypoint Mobility Model

In performance studies of wireless systems, e.g., ad hoc networks, the movement of the users must be modelled. In practice simple synthetic mobility models are often used, since more realistic models are not readily available. Random Waypoint model (RWP) is one of the most widely used synthetic mobility models. In this model, a mobile node basically moves along a zigzag line, where each of the straight line segments is called *a leg*. At each turning point the node chooses a new destination randomly (from uniform distribution in the given area $\mathcal{A}$)and then moves towards the destination at a constant speed, which can be drawn from a given speed distribution at each turning point. The node may also remain stationary for a random think time before starting its movement towards the new destination.

To be exact, the process representing the movement of a node within the simulation area according to the RWP model can be described as follows. Initially, the node is placed into the area at the point P_1 according to a uniform distribution over the simulation area $\mathcal{A}$. Then a destination point (also called waypoint) P_2 is chosen from a uniform distribution and the node moves along a straight line from P_1 to P_2 with constant velocity V_1 drawn independently of the location from a distribution $f_V(x)$. Once the node reaches P_2, a new destination point, P_3, is drawn independently from a uniform distribution over the simulation area and velocity V_2 is drawn from $f_V(x)$ independently of the location and V_1. The node again moves at constant velocity V_2 to the point P_3, and the process repeats. This is illustrated in Figure 2.1.

The simulation area $\mathcal{A}$ can be any convex area. However, in this exercise we will be studying the RWP process in the unit disk. Furthermore, the velocity distributions to be used are: constant velocity and uniformly distributed velocity in the range $[v_{\min}, v_{\max}]$.

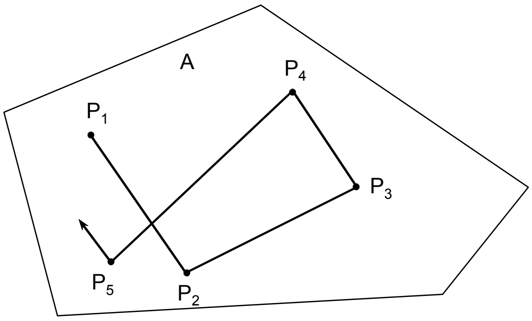

Figure 2.1: Zigzag movement of the RWP process.

2.2 Connectivity

In wireless ad hoc network two nodes can communicate directly with each other, if their *transmission radius* is greater than the distance between them. The network is *connected*, if there is a path from every node to every other node. Connectivity is an essential parameter for performance of communication networks, since it tells us whether we can send data to each of the destination nodes. In this exercise, we will study both the probability that the network is connected, i.e., the proportion of time the network is connected, and the mean duration of the connectivity period.

In this exercise we assume that each node has the same maximum transmission radius, which is denoted by d. Then, at an arbitrary time instant, if the distance between two nodes is shorter than the transmission radius, these two nodes are connected with a link. The connectivity of the whole network can be examined for example by the breadth-first search algorithm.

2.3 Simulation of connectivity measures

To study connectivity via discrete event simulation, the simulation is carried out by examining the system state (connected/disconnected) between fixed time intervals of length Δ. Thus, the continuous time system will become discretized, but the error from this can be made arbitrarily small by keeping Δ sufficiently small.

Consider a simulation with k nodes with the simulation time being equal to $N \cdot \Delta$, and let C_n denote the boolean random variable for weather the network was connected or not during the nth interval,

$$C_n = \begin{cases} 1, & \text{if network is connected at } n\text{th interval,} \\ 0, & \text{otherwise.} \end{cases}$$

Probability of connectivity: Given the number of nodes k and the transmission radius d, the probability of connectivity, $c_{k,d}$ equals the fraction of time the network is connected, and the estimator for it equals

$$\hat{c}_{k,d} = \frac{1}{N \cdot \Delta} \sum_{n=1}^{N} C_n \cdot \Delta = \frac{1}{N} \sum_{n=1}^{N} C_n.$$

Mean length of connectivity periods: To estimate the mean length of the connectivity periods, we need to keep track of "how long" the periods of consecutive connectivity (and disconnectivity) periods are. Let us denote by $\bar{T}_{k,d}$ the mean connectivity length with k nodes, each with a transmission radius d. In the simulation, let T_n denote the length of the nth connectivity period. Thus, the measured variables T_n represent the lengths of the continuous sequences of C_i:s for which $C_i = 1$. For example, assume that $T_2 = 5 \cdot \Delta$. This means that in the simulation the second connectivity period consisted of five consecutive sampling periods where the network was connected until it became disconnected again. Thus, the estimator for $\bar{T}_{k,d}$ is simply

$$\hat{T}_{k,d} = \frac{1}{N} \sum_{n=1}^{N} T_n,$$

where N is the total number of connectivity periods observed during the simulation. Note that one actually only needs to record the counts for the connectivity lengths, since we are sampling at fixed intervals.

Chapter 3

Exercise

3.1 Practical Instructions

CNCL is installed in all workstations in computer class B215. Remote login to the workstations is also possible, see the list of workstations from

> **http://www.ee.hut.fi/unix/hardware.shtml**.

In order to be able to do the simulation exercise, you have to go through the following steps:

1. Copy the necessary files (cnclexer.c and Makefile) from the course webpage to your working directory.
2. If your default shell is tcsh, write a shell script including the following lines:

   ```
   setenv LD_LIBRARY_PATH /usr/lib:/usr/local/lib:$LD_LIBRARY_PATH
   setenv PATH /opt/csw/gcc2/bin:$PATH
   ```

   If your default shell is bash, write a shell script with the following:

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH: /usr/lib:/usr/local/lib
   export LD_LIBRARY_PATH
   PATH=/opt/csw/gcc2/bin:$PATH
   export PATH
   ```

   Execute the script with the command “source usefile", where usefile is the name of your shell script. By default your shell should be tcsh and these commands should work (you can check your shell type with the command “echo $SHELL")

3. Edit the source file cnclexer.c: Implement the required methods and the main program.
4. When you want to compile your program, create first a .depend file with the command “touch .depend".
5. Compile the program by “make”.
6. Execute the program with the command “./cnclexer".

If you need to look at the *.h or *.c files of CNCL classes, the CNCL library is installed at:

/p/gen/courses/S38/S38.148/cncl/src/cncl-2.1/

The *.h files are under include-directory and *.c files under lib-directory.

3.2 General description of the task

In this exercise the task is to:

1. Implement the RWP mobility model. This first part is simply about this and you need to verify that it works.
2. Study the connectivity properties of the network for varying number of nodes and different speed distributions.

3.2.1 Description of the simulator objects

- **Controller** Object contains methods that are used for calculating if the network is connected at a certain time instant. This object controls the program execution and data collection.
- **Node** Object has methods for movement handling.

3.2.2 Description of the simulator events

In CNCL, objects communicate by sending events to each other. These events will then be handled by the **event_handler()** method of the object. In this exercise the Controller and Node classes can handle events. The used event types are the following:

1. **EV_TIME** Controller's event to itself every t seconds, where t is a fixed sampling time. The state of the system (connected or not) is studied.
2. **EV_NODE** Node's event to itself when the node arrives to destination point. A new destination point and a new speed are drawn from suitable distributions.

3.3 Detailed task description and simulation parameters

Below are some fixed system parameters:

- All simulations correspond to steady-state simulation. However, the system, i.e., moving RWP node(s), is not in the steady state at the beginning of the simulation. Implement initial transient control and start data collection after 10 time units.
- The system state is sampled at fixed intervals. The sampling interval (**EV_TIME**) should be less than 0.01 time units.
- The simulation area is the *unit circle*, i.e., circle with radius $r = 1$. Thus, when an RWP node arrives at the next way point, the next way point is chosen from the uniform distribution within a unit circle. To generate these points you can use a simple rejection method or use the polar coordinate method in problem 1 from exercise 4.

3.3.1 Part 1

The first task is basically to just implement logic of the movement of the RWP node and verify that it works properly by performing the following:

1. Simulate 1 node with a constant velocity $v = 1$. Estimate the mean distance of the node from the center of the circle, when the node location is sampled every **EV_TIME** time units. The exact answer should be 0.516.
2. Again simulate 1 node but with a random velocity distribution so that $v \sim U(0.1, 1.9)$, i.e., the velocity of the node is uniformly distributed between 0.1 and 1.9 and $\mathrm{E}[v] = 1$. Now measure the velocity of the node, when the node is sampled every **EV_TIME** time units. What is the mean value of your velocity samples? It represents the mean value of the stationary distribution of the node velocity. Can you explain why the value is not just simply $\mathrm{E}[v] = 1$.

Deadline: The deadline is the 2nd question hour on week 46, i.e., Thu, 15.11.2007, at 10:00.

Handout: Just a printout from your simulation that demonstrates your results for the tasks 1 and 2 above. Show it to the assistant at the 2nd question hour or send it via e-mail to: tuomas@netlab.tkk.fi.

3.3.2 Part 2

Here we study the connectivity properties as a function of the transmission radius for $n = \{2, 10\}$ and two velocity distributions. More specifically, the tasks are:

- **Case1:** Use a constant velocity $v = 1$ and number of nodes $n = 2$. Estimate the connectivity probability for following values of the transmission radius, $r = \{0.25, 0.5, 1.0, 2.0\}$. Also, estimate the mean duration of the connectivity periods for following values of the transmission radius, $r = \{0.25, 0.5, 1.0\}$. Perform the same simulation when number of nodes $n = 10$.
- **Case2:** Velocity is drawn from a uniform distribution $\sim U(0.1, 1.9)$, average speed $\bar{v} = 1$. Repeat the same simulations as above.

In your final report, present your results either in the form of graphs or tables. Also remember to include 95% confidence intervals (either based on independent replications or batch means method, **describe your method in the report!**). In the report, discuss the following issues: How does the transmission radius affect the connectivity? How does the number of nodes affect the connectivity? Does the velocity distribution have an effect on the mean duration of the connectivity period? Does the velocity distribution have an effect on the probability of connectivity?

3.3.3 Handout requirements

The following items should be included in your final report:

- All source code with comments
- Graphs of the simulation results
- Brief analysis of the results

3.3.4 Returning

The deadline of part 2 of the exercise is Monday, 26.11.2007 at 24:00 o'clock. Return your report and codes via e-mail to: tuomas@netlab.tkk.fi.

3.4 Getting help

The students are **strongly** encouraged to attend the two exercise classes that are organized as “question hours" in the computer class room B215. The idea is that the assistant will be present there to answer your questions and maybe even to help in solving some specific problems. The dates for the question hours dealing with CNCL are:

- Friday, 9.11.2007, at 10-12, place B215
- Thursday, 15.11.2007, at 10-12, place B215

In case the above is not enough, contact the assistant (tuomas@netlab.tkk.fi).