

## Assignment 3: Distance-Vector Routing

*Due date: December 21, 2018 at 22:00*

*This is an individual assignment. You may discuss it with others, but your code and documentation must be written on your own.*

### Problem

The goal of this assignment is to implement a *distance-vector* routing protocol within the *SimpleNet* network simulator, which is available from the course Web page along with documentation and examples. Your implementation must consist of a class called *DVRouter* that extends the *simplenet.Router* class. You may define and use other classes. In particular, you will certainly need to define some classes that represent the messages exchanged by routers. Your implementation must cover all the fundamental aspects of a distance-vector routing protocol. In particular, (1) a router must maintain and update its distance vector and its next-hop vector; (2) a router must communicate its distance vector to its neighbors; (3) a router must define the appropriate forwarding table entries using its next-hop vector.

You may assume that the network is perfectly reliable and immutable. So, your router does not have to update routing information proactively.

### Suggestions

You should first familiarize yourself with the *SimpleNet* simulator. You should do that by compiling and running the examples available on-line. First, compile one of the example routers, then run it with one of the example workloads. After that, create other workloads, possibly by modifying the available workloads, and test the router with your new workloads. After that, try to modify the example implementation. And again, test your modified version with the available workloads. Once you are familiar with *SimpleNet*, start working on your implementation of *DVRouter*. Feel free to use one of the examples as a skeleton. But most importantly, you should use the description of distance-vector routing given in class, together with the one available in the book, as a specification for your program.

### Submission Instructions

Submit all your source files. Add comments to your code to explain sections of the code that might not be clear. You may use an integrated development environment (IDE) of your choice. However, *do not submit any IDE-specific file*, such as project description files, and *make absolutely sure that the files you submit can be compiled and tested with a simple invocation of the standard javac compiler and the standard java virtual machine*.

In addition to the source files, submit a text file called *README* containing a brief description of your implementation, including a description of the communication protocol you developed, and also a list of all the limitations and errors you are aware of were unable to fix.

Package all the files you need to submit in an archive file named

`a03-<lastname>-<firstname>`

and submit that file through the iCorsi system.