

CS 5480/6480: Computer Networks – Spring 2012
Programming Assignment 2
Due by 11:59:59 PM MT, April 2nd 2012

Important:

- **No cheating will be tolerated.**
- **No late submissions.**

Total Points for this programming assignment: 100

The goal of your programming assignment is to build and experiment with Distance Vector Routing Protocol (DVRP). In this assignment all nodes must be implemented on the same machine. Each node is identified by one or more port numbers.

1. Implement DVRP on top of UDP sockets to demonstrate the count-to-infinity problem with three nodes.
2. Add Poisoned-reverse to your code to show that the count-to-infinity problem is solved in the three node example used in the first part.
3. Construct a four node example to show that the count-to-infinity problem exists even after adding Poisoned-reverse.

To implement DVRP you need to define data structures for the distance vectors and the distance vector table. You are expected to keep the implementation simple and stay focused on solving the above three problems.

Note: You can start DVRP from a stable initial condition. You do not have to run the DVRP algorithm to reach an initial stable condition. In the second and the third part of the question you can also assume that the right Poisoned-reverse information is available at the nodes to begin with. What I would like to see is the DVRP message exchange and the changes in Distance vectors in the three cases when the link cost changes. The “right Poisoned-reverse information is available” means that the nodes have been informed about poisoned-reverse distances and your program does not have to worry about making the network nodes learn this information by exchanging messages.

Grading Policy

The program(s) you hand in should work correctly and must be well documented. You should submit your programming assignment *electronically* using the *handin* command. Your submission should include the following:

1. The entire code containing in-line documentation.
2. A separate document of a page or two (at most) describing the solution separately for DVRP, overall program design, a verbal description of “how it works”, and design tradeoffs considered and made. Also describe possible improvements and extensions to your program (and sketch how they might be made).

3. A separate description of the tests you ran on your programs to convince yourself that they are indeed correct.
4. **A plain text *readmeDVRP.txt* file describing how to run your programs.**
5. The executable files of your programs.
6. A plain text files *outputDVRP.txt* containing sample outputs as required by the assignment.

Grading

Program Listing

works correctly	80%
in-line documentation	5%
exception handling	5%

Design Document 5%

Thoroughness of test cases 5%

Electronic Submission - Turning in files using handin in the CADE

All programming assignments must be submitted on CADE machines using the *handin* command. To electronically submit files while logged in to a CADE machine, use:

```
% handin cs5480 assignment_name file_1 file_2 ...
```

cs5480 is the name of the class account (same for both *cs5480* and *cs6480* students), and assignment name (*PA1*, *PA2*, or *PA3*) is the name of the appropriate subdirectory in the *handin* directory. Use *PA2* for this assignment.

Some other important points

- Every programming assignment of this course needs to be individually done by a student. No teaming or pairing is allowed.
- You need to program in one of the following three languages: C, C++, or Java
- **Please do your assignment on CADE Lab machines only.**
- **Use port numbers in the range 6000 – 8000.**
- *readmeDVRP.txt* – This file must contain compilation and running instructions detailing how the TA should go about running your programs (are there any specific addresses/port numbers you have hard-coded, is there any specific order in which the executables must be executed, etc).