

ANC4: Assessed Exercise 2013

Deadline: Friday 8th March (9am)

Weighting 20%

Description

Consider a simplified point-point network description consisting of a set of nodes {N1, N2, ...} and a set of links with costs, of the form (N1,N2,C). In addition, assume every node has a routing table with 3 columns: destination, distance and outgoing link.

With a programming language of your choice* implement an application which will read a **text-file** containing a simple human-readable description of such a network and then simulate distance-vector routing with synchronised periodic exchanges, updating all node routing tables over any selected number of iterations.

A user should be able to:

- compute routing tables for any preset number of exchanges or until stability is achieved;
- preset any link to change cost or fail after any chosen exchange (you may assume for simplicity that neighbours notice cost changes or unreachable neighbours immediately);
- view the best route between any source and destination after any chosen iteration
- trace the routing tables of any set of nodes for any specified number of iterations in a way that can be easily specified and viewed;
- engage, **optionally**, a split-horizon capability to help combat slow convergence
- run the application under Windows 7 x64 (a **good, easy-to-use** command line interface is OK).

Once your simulator is working, document two simple examples with appropriate input networks to illustrate how some exemplar routes and tables evolve under:

- normal convergence of the distance-vector algorithm;
- slow convergence both **with and without** the split horizon facility switched on.

Describe clearly but briefly, with the help output from your application, what is going on as the routes evolve. Use diagrams where appropriate to aid your description.

Deliverables.

Submit the following in a zip or rar file attachment via email to lewis@dcs.gla.ac.uk by Friday 8th March. The subject heading in your email MUST be: "ANC4 assessed exercise".

1. A short (< 1 page) informal overview of your high-level application design and a short user manual (< 1 page) including instructions to install and run the program*. (~10%+10%).
2. A well-commented source listing (informative commenting is what counts). Give clear instructions on how to build the executable. (~10%).
3. A functioning executable file which MUST be the compiled version of 2. Marks are gained for implementing the required functionalities and for an application which is easy to use. A graphical interface is not necessary (although you may provide one if you wish) (~30%).
4. For your two example networks, give input description files and, using words, diagrams and output from your application, a documented "walkthrough" of how some exemplar tables and routes evolve as each iteration completes. Marks here will be awarded for **clarity and brevity**. (~30%).
5. A (1 page max) status report discussing any limitations of your program and how you would improve it given the time. (~10%).

All documents must be **either in MS Word or Acrobat PDF format**. Marks will be awarded for **clarity and brevity** of presentation. If you do not provide adequate instructions for building or use of your application, or if your output is hard to interpret, you will be marked down.

Items 1, 2, 4 and 5 should also be submitted in hard copy in the box provided in Lilybank Gardens.

* **NOTE:** Although you may use any programming language you like, you must ensure your executable will **run** without requiring the user to install any additional commercial software. If **ANYTHING** (e.g a JVM) is needed which would not be present on a freshly installed copy of Windows 7 x64, you **MUST** include instructions about how to get it and install it!