CNPro: Computer Networks Project

A toy DV/RIP simulator in Java

This project is an extra-credit homework submission, it consists of extending a provided base Java code towards building a RIP basic simulator (Routing protocol based on the Bellman-Ford algorithm).

Goal

The goal of this homework project consists of reviewing the concepts and basic workings of the Routing Information Protocol, specifically we aim towards simulating the Distance Vector algorithm in Java. CNPro is not a mandatory homework since it only contributes to your passed mark in the form of extra credit.

The requirements for the software

This project consists of extending a provided base Java code towards a DV algorithm basic simulator. According to the functions of this Routing Protocol as stated in chapter 3 of Peterson & Davie textbook on Computer Networks, you are required to provide the following functionalities in your simulator:

1. Each router is simulated in a standalone instance of the simulator (A RIPSim class is included with the provided basic Java classes)
2. Each router listens on its UDP socket (Local node's IP and a UDP port higher than 50000)
3. The network graph notion of connectivity is captured in the simulator by the sending of DV's over a node's UDP socket to a known neighbor. A router is connected to another router by knowing its IP/PORT address and exchanging DV's with it according to the Bellman-Ford formula and algorithm
4. The classes provided contain examples that illustrate how to represent a routing entry (A DV), how to create timer tasks in Java so that the simulator can schedule the reception and updating of neighboring routers
5. Each router (RIPSim running instance) will print its DV periodically in order for the user to become aware of the behavior of the algorithm, each RIPSim instance is to be run on a separate shell window
6. You may want to create more timers and schedulers so that your simulator faithfully represents the DV algorithm
7. The basic network topology of Fig.1 will suffice for this project, for the time being, we are not seeking a simulator that can simulate any network graph
8. Alongside with your commented source code, you must provide a short writeup explaining your understanding of the DV algorithm, your software design decisions, particularly the main software entities and data structures and your tests. Writeups properly composed in English will receive a linear additional 0,5 grade points.

9. HOMEWORK SUBMISSION DEADLINE: Compress your sources by using the zip/gzip compressing formats ONLY alongside with your writeup and send it to foces.informatica.unileon@gmail.com by Saturday 22/june/2013.
10. CNPro contributes a maximum of 2 grade points according to the degree of completion of the simulator and the quality and originality of your code. Collaboration is permitted and encouraged at the conceptual level, however, your code and documentation must be original with no exception whatsoever. Contact me if you need any assistance, either by mail or in person in my office.

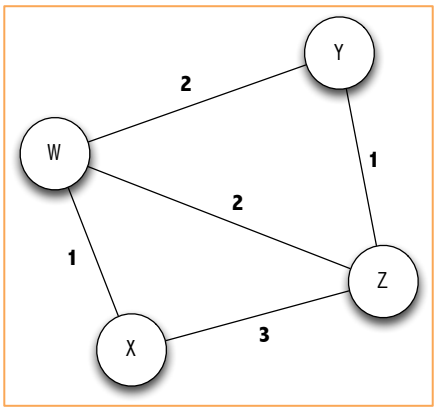

Fig. 1. Network topology for CNPro RIP simulator

Comments on the source code

The base source can be downloaded from http://paloalto.unileon.es/cn/cnpro.zip, it is commented and its goal consists of offering you a base code so that you do not have to start from scratch. It contains three classes, the main simulator class, an example timer task class and an example basic routing table entry. The most straightforward way for getting started consists of running the provided program, observing its behavior, trying to conceive its relationship with the DV algorithm and performing small, incremental changes and test as much as you can as you write —*and play with the code.* You need a single machine for running the topology mentioned above, recall that each router is represented by a Java instance of RIPSim and that each instance should establish a Java UDP socket on the host's IP address and a UDP port higher than 50000 (By convention). If you can run the simulator on several machines, the better for improving your sockets programming and network skills, but it is not strictly necessary.