Programming Project 1: CS 6390

Ravi Prakash

Summer 2020

1 Assignment

Refer to slides 10-12 on RIP, describing the count to infinity problem, and the *split horizon* and *split horizon with poison reverse* solutions. In this project, you are required to simulate the three situations for the example shown in slide 10. There are five routers, A through E, arranged in a line, with link costs of 1, and router A fails.

Information exchange and state change happens in a synchronous fashion as described below:

1. In the beginning, *i.e.*, prior to round 1, the routing tables of B, C, D and E have their distance to router A as 1, 2, 3, and 4, respectively, and each router knows who its neighbors are.
2. Router A fails just before round 1.
3. In the beginning of each round, each router sends its distance vector to all its neighbors. The communication channel between neighboring routes is reliable. So, each message sent to a node by its neighbor in a given round is delivered to the node before the end of the same round.
4. At the end of a round, each router updates its distance vector based on its previous state and the distance vectors received from neighbors.

Show how the routing table of each router changes with each round, especially with respect to the entry for router A, until the distance estimate for A reaches infinity for the following scenarios:

1. Basic distance vector exchange between routers,
2. Distance vector exchange with split horizon, and
3. Distance vector exchange with split horizon and poison reverse.

2 Requirements

1. Source code must be in the C/C++/Java programming language.
2. The program must run on UTD lab machines (`dc01, dc02, ..., dc45`).
3. You will need to know socket programming and its APIs for the language you choose. It can be assumed that each router is running on its own machine (`dcXY`) and has secure socket connections with its neighboring routers. Please get familiar with basic UNIX commands to run your program on *dcXY*.

3 Submission Information

The submission should be through eLearning in the form of an archive consisting of:

1. File(s) containing the source code. Your source code must have the following, otherwise you will lose points:
   (a) Proper comments indicating what is being done.
   (b) Error checking for all function and system calls.
2. The README file, which describes how to run your program.