CPS706 Group Project

Winter 2023

Objective:

The objective of this project is to build an educational interactive tool that will help students in the Computer Networks course better understand routing algorithms. Your tool should implement two routing algorithms (details are given below), permit the user to modify input parameters, and provide students with an enhanced visualization experience via a user-friendly interface.

Routing Algorithms:

The goal of routing algorithms is to determine good paths (equivalently, routes), from senders to receivers, through the network of routers. Typically, a "good" path is one that has the least cost.

One way in which we can classify routing algorithms is according to whether they are centralized or decentralized.

1. A centralized routing algorithm computes the least-cost path between a source and destination using complete, global knowledge about the network. That is, the algorithm takes the connectivity between all nodes and all link costs as inputs. This then requires that the algorithm somehow obtain this information before actually performing the calculation. Algorithms with global state information are often referred to as link-state (LS) algorithms since the algorithm must be aware of the cost of each link in the network. The link-state routing algorithm that is widely used is known as Dijkstra's algorithm, named after its inventor.
2. In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner by the routers. No node has complete information about the costs of all network links. Instead, each node begins with only the knowledge of the costs of its own directly attached links. Then, through an iterative process of calculation and exchange of information with its neighbouring nodes, a node gradually calculates the least-cost path to a destination or set of destinations. A well-known decentralized routing algorithm is called a distance-vector (DV) algorithm because each node maintains a vector of estimates of the costs (distances) to all other nodes in the network. This algorithm relies on the Bellman-Ford equation to calculate the least costs of paths.

What to hand in:

Upload the following to D2L:

- Installation manual
- Recorded demo with explanation (10 minutes maximum)
- Source code (Zip file)

Make sure you include your names and student IDs on the title page of your installation manual

Group Size:

- 5-6 students (no exceptions). Note: you may work with students in different sections as long as you have the same TA

Deadline:

- Live demo: during the last week of lab sessions. All group members are expected to attend and must be ready to answer any questions related to their project.
- D2L submission: April 14, 11:59 PM. Late submissions will not be accepted!

Grading Rubrics:

- Implementation of algorithms and requirements (50%)
- User interface (20%)
- Coding style (10%)
- Creativity (20%)