CS 513 Project 4

Network Layer Routing

Proposal Due: November 5, 2025

Project Due: December 12, 2025

Team: Lord of the Pings

- Alexander Beck
- Jack Rothenberg

Project Design

For this project, we will be developing in Python 3, utilizing git for code management. The code itself will be ran in a large loop, checking for user input commands to properly process itself. It will check for different inputs from the command line including for constructing/modifying the graph, along with running/analyzing the different algorithms mentioned below.

Graph Representation

We will be storing all the graph nodes and edges in the form of a dictionary. Helper functions will be created as necessary to accomplish the following:

- Check if a node exists
- Check if an edge exists
- Get immediate neighbors for a given node
- Get cost of an edge
- Update cost of an edge
- Remove edge
- Remove node
- Add node
- Add edge
- Print out the graph (optional)

Link-State Routing Algorithm

Run Dijkstra on the graph with the inputted node as the root and store the resulting spanning tree as a routing table. Also make the shortest path algorithm swappable to compare results and more easily run experiments.

Distance-Vector Routing Algorithm

Run an algorithm to calculate the distance to each node's neighbors, and share these distances to other nodes. Combine the distances in each node's routing table and share this routing table with other nodes. This is ran one iteration per command line call, and can be repeated until the distance values converge (the shortest path was found).

Tentative Schedule

Week 1 (11/04/2025):

- Project proposal submission (Alex/Jack)

Week 2 (11/11/2025):

- Project proposal revising (if needed) (Alex/Jack)
- Complete graph input and creation (Alex)
  - From command line
  - From file
- Create and test Link-State Routing algorithm (Jack)
  - Command line inputs
  - Print routing table for specified node

Week 3 (11/18/2025):

- Create and test Distance-Vector Routing Algorithm (Alex)
  - Including command line inputs
  - Print routing table for specified node

Week 4 (11/25/2025):

- No scheduled work
  - Thanksgiving Break

Week 5 (12/02/2025):

- Create different scenarios to compare the two routing algorithms and analyze results (Alex/Jack)
- Time permitting, add additional features (optional)
  - Command to show intermediate results from all algorithms
  - Command to visualize constructed network graph
  - Command to visualize resulting spanning tree
- Start writing report (Alex/Jack)

Week 6 (12/12/2025):

- Finish & submit report (Alex/Jack)