Shortest Path Routing using Dijkstra Algorithm

SUBMITTED BY

Hosni Adel Hosni Abdel Megged	1700455
Hassan Mohamed Abdel Salam	1700454
Hussien Mostafa Zaky	1700463

SUPERVISED BY

Dr. Hossam Fahmy

Cairo, 2021

Introduction & Background

Routers use routing algorithms to find the best route to a destination. When we say "best route," we consider parameters like the number of **hops** (the trip a packet takes from one router or intermediate point to another in the network), time delay and communication cost of packet transmission. We will use dijkstra's shortest path algorithm to find the best route to a destination.

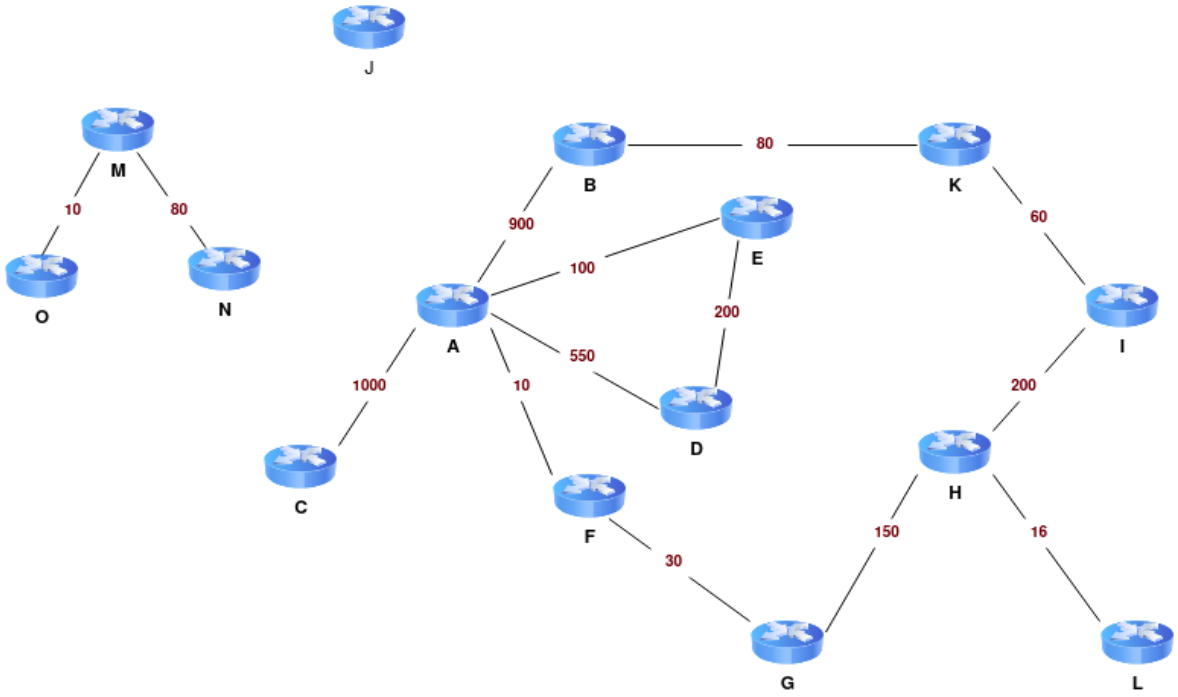
dijkstra's shortest path is an algorithm created by Dr. Edsger Dijkstra for calculating the shortest path between vertices in a weighted graph

Dijkstra's shortest path applications

- Road networks
- Electricity lines
- Google maps
- Social networking apps
- Flighting agenda
- Ip routing

Demo

Assume we have the following network and we need to get the shortest path between Router A and the other routers in the network:



In the input.txt file we enter the number of nodes and represent the pervious network:

```
15
0 1 900
0 2 1000
0 3 550
0 5 10
5 6 30
6 7 150
7 8 200
7 11 16
8 10 60
1 10 80
12 14 10
12 13 80
3 4 200
```

Build and the run dijkstra algorithm on our nodes and the output will be:

```
hosni adel
hosni@hosniadel:~/A:Hosni/4th CSE/Networks/assignment-II/dijsktra$ g++ main.cpp -o main.out
hosni@hosniadel:~/A:Hosni/4th CSE/Networks/assignment-II/dijsktra$ ./main.out
Cost from router A to router A is 0
Cost from router A to router B is 530
Cost from router A to router C is 1000
Cost from router A to router D is 550
Cost from router A to router E is 750
Cost from router A to router F is 10
Cost from router A to router G is 40
Cost from router A to router H is 190
Cost from router A to router I is 390
Impossible router A to reach router J
Cost from router A to router K is 450
Cost from router A to router L is 206
Impossible router A to reach router M
Impossible router A to reach router N
Impossible router A to reach router O
hosni@hosniadel:~/A:Hosni/4th CSE/Networks/assignment-II/dijsktra$
```

Source Code

```cpp
#include <iostream>
#include <vector>
#include <queue>
#include <sstream>
#include <fstream>
#include <map>

using namespace std;

vector<int> cost;
vector<vector<pair<int, int>>> v;
map<int, char> m;

void apply_dijsktra(int nodes);
int load_input(std::string path);

int main()
{
  int nodes = load_input("./input.txt");

  apply_dijsktra(nodes);

  for (int i = 0; i < 15; i++)
  {
    if (cost[i] == 1e9)
    {
      cout << "\033[1;31"
           << "mImpossible router " << m[0] << " to reach router " << m[i]
           << "\033[0m\n";
    }
    else
    {

      cout << "Cost from router " << m[0] << " to router " << m[i] << " is " << cost[i] << endl;
    }
  }
  return 0;
}
```

```cpp
int load_input(std::string path)
{
  int nodes;
  std::fstream newfile;
  std::string input;

  newfile.open(path, std::ios::in);
  if (newfile.is_open())
  {
    getline(newfile, input);
    nodes = stoi(input);
    v.clear();
    v.resize(nodes);

    for (int i = 0; i < nodes; i++)
    {
      m[i] = 'A' + i;
    }

    while (getline(newfile, input))
    {
      int arr[3], index = 0;

      std::string word;
      std::stringstream iss(input);

      while (iss >> word)
      {
        arr[index] = stoi(word);
        index++;
      }

      v[arr[0]].push_back(make_pair(arr[1], arr[2]));
      v[arr[1]].push_back(make_pair(arr[0], arr[2]));
    }

    newfile.close();
  }
  return nodes;
}

void apply_dijsktra(int nodes)
{
  priority_queue<pair<int, int>> q;
  q.push(make_pair(0, 0));

  for (int i = 1; i <= nodes; i++)
    cost.push_back(1e9);

  cost[0] = 0;
  while (q.size())
  {
    int cst = -1 * q.top().first, u = q.top().second;
    q.pop();

    if (cost[u] < cst)
      continue;

    for (int i = 0; i < v[u].size(); i++)
    {
      int d = v[u][i].first;
      int c = v[u][i].second;
      if (cst + c < cost[d])
      {
        cost[d] = cst + c;
        q.push(make_pair(-cost[d], d));
      }
    }
  }
}

```

Repository

link: https://github.com/hosniadel666/shortest-path-routing.git