Dynamic Routing in a Mesh Network

Emphasis on Throughput

Nick Alvarez and Jayam Sutariya
Department of Computer Science and Engineering
University of Nevada, Reno
7 December 2020

Introduction

Packets must travel to their destination in an optimal way, accounting for possible congestion at a given router. They will travel along a specific path of nodes, considering the shortest distance and the lowest delay at each router. The goal is to create an algorithm that will route packets through the network in this optimal way, avoiding congestion to maximize throughput.

Functionality

Documentation of the code can be found in the Documentation file or online at the project GitHub repository.

Initially, the team had to create a network of routers to work with. The team decided to create manually generated network instead of a random one to compare results with different realistic scenarios. The team went through a couple of designs for the problem and finally landed on the network shown in the figure below. The reason the team picked this specific design is because this network results in a lot of different paths that can be taken after the addition of realistic aspects. Furthermore, all the distances between the nodes are the same – 1.

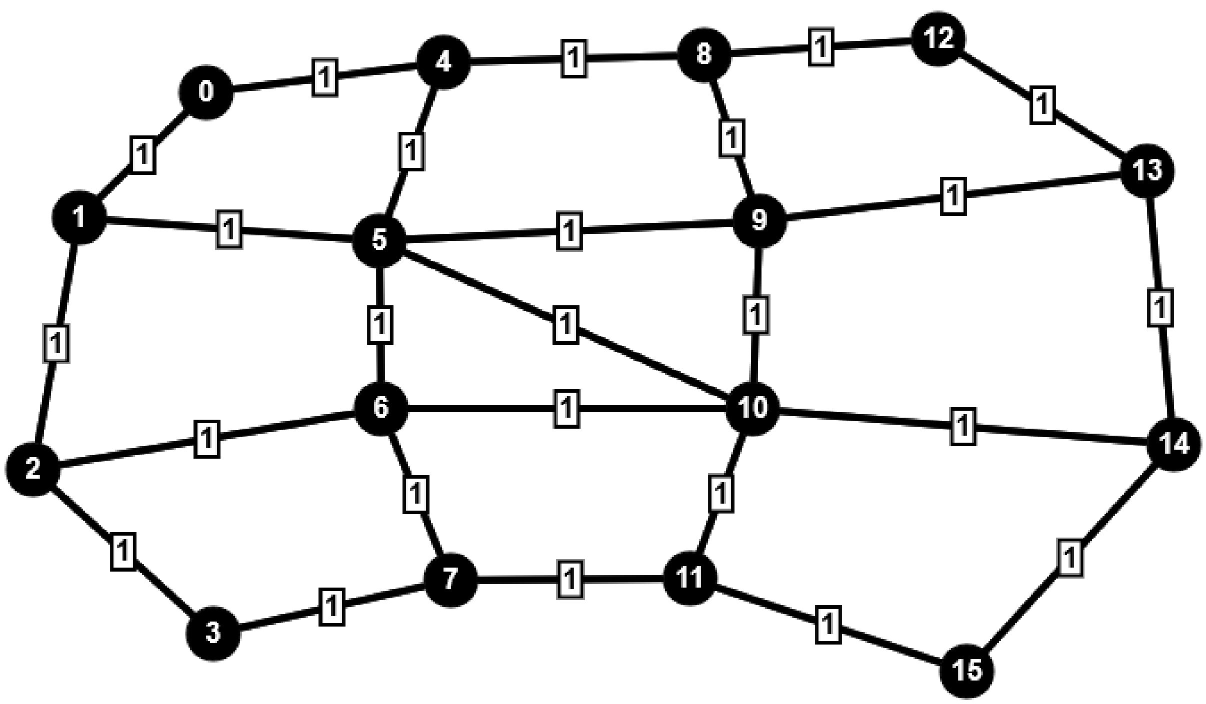

Fig. 1: The network graph. All distances between adjacent nodes are one.

For all the simulations, the team picked for a packet to traverse from node 0 to node 15. As seen from the figure below, the shortest path is highlighted. If one simply runs Dijkstra's shortest path algorithm on this specific network, it will always result in the red path.

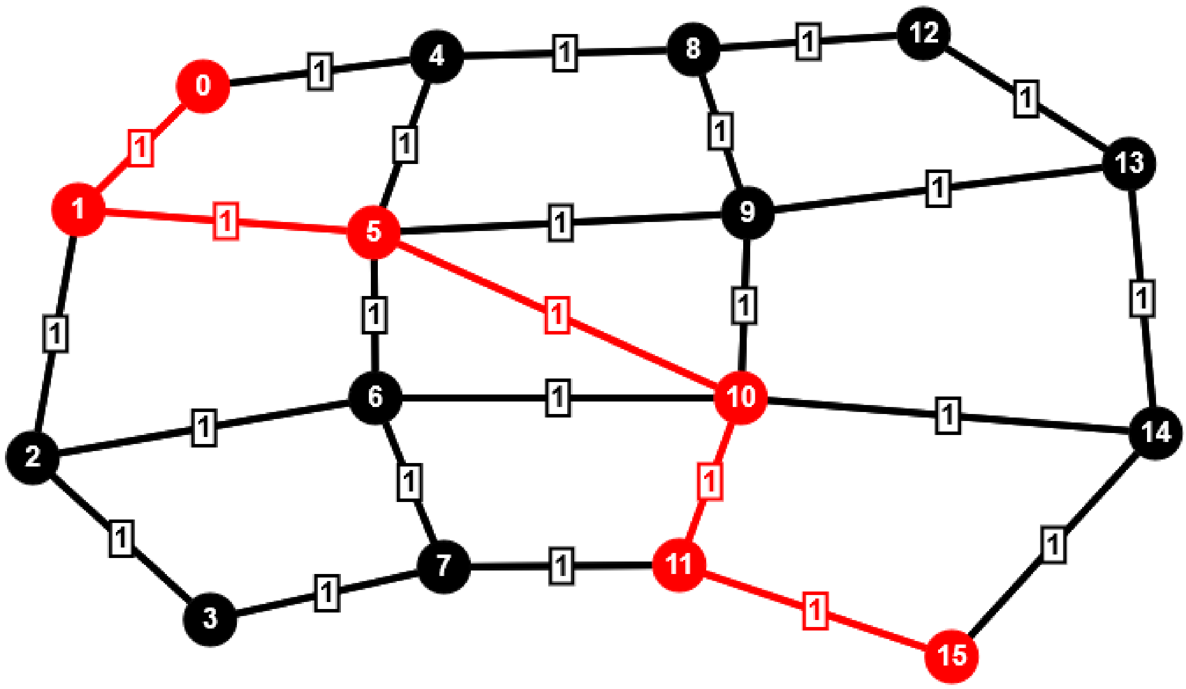

Fig. 2: Dijkstra's algorithm performed from node 0 to node 15. This is the shortest path on the graph for these two nodes, every time, if the router chooses a path strictly based on distance.

Now that the foundations are set, the algorithm starts by randomly assigning nodal processing delay and queuing delay, which are Router class private variables, to each router before each packet is about to be sent. At this moment, the shortest path algorithm will not only consider the distance, but also the delays associated with each node to represent congestion, this being a realistic aspect of the simulation. Moreover, each time a packet is about to be sent, new delays are set to the routers as how it would be in reality. In addition, there is also a random chance that the packet may be lost as it does happen. After all these realistic considerations, the algorithm displays the most efficient path every time a packet is sent and displays the time it took to travel and whether the packet was lost or not.

To make the simulation more realistic and abstract, more nodes can be added or even a completely different network of routers can be created easily, however, that must be done manually. The simulation first asks the number of packets to be sent, further increasing the realistic aspect. The next chronological functionality of the simulation is to ask the user origin and the start and end point of the packet traversal, further adding onto the realistic aspect. The next functionality of the simulation is that it asks the user if they want a detailed version of the analysis or not.

Novel Contributions

Arbitrary Number of Packets

One of the functionalities of the simulation is that it allows the user to pick how many packets they desire to send. The more packets sent, the more variation you can see in the output. The time required to send the packets will be different each time due to the randomness. Thus, replicating a realistic scenario.

```
Change number of packets to send? Default 1. (y/n): y
Packets to send: 5
```

Fig. 3: If the user chooses, they can change the number of packets that are sent.

Verbose Mode

The simulation also allows the user to enable verbose mode. Normally, the simulation would only output the time it took for a certain packet to traverse and whether or not if that certain packet was lost as seen in the figure below.

```
#        Time             Lost
----------------------------
1        13.295           0
2        26.2             1
3        16.264           1
4        20.082           0
5        21.06            0
```

Fig. 4: Verbose mode is disabled, listing only the packet number, time of travel, and number of lost packets.

By enabling verbose mode, the simulation lets the user analyze the results more deeply. Additional details are displayed regarding the packet traversal. Those additional details include exactly where the packet loss happened, the exact path that the packet took and the usual packet travel time and whether packet loss happened or not.

```
Lost packet! From 9 to 13.
0 -> 1 -> 5 -> 9 -> 13 -> 14 -> 15
Travel time: 20.8383 ms
1 lost packet(s)
```

Fig. 5: Verbose mode is enabled, listing any lost packets and their intended route, the final path of the packet, total travel time, and number of lost packets.

Packet Loss

Another important novel functionality that this simulation provides is the presence of packet loss. To replicate the realistic element of packet loss, the algorithm sometimes randomly chooses to drop a packet each time a packet is sent between nodes. Furthermore, the algorithm also displays between which two nodes the packet loss happened.

<u>Congestion Simulation</u>

To replicate realistic situations, the team added the concept of congestion to the network. The algorithm randomly applies processing and queuing delays when a packet is being sent. Thus, the congestion is modeled, and the algorithm takes a path based on the least congestion. Moreover, the algorithm applies such delays every single time a packet is about to be sent resulting in a more realistic case where the congestion changes over even the smallest increments of time.

Analysis of Results

The program runs as expected and provides a good look at the routing decisions in a congested network. Testing with multiple packets, some packet loss does occur, as expected. There are variations in travel time for each packet, even with the same path and zero packet loss, due to the nature of the congestion simulation for each router.

```
##### Dynamic Routing Simulator #####
The network will be generated with 16 nodes.
Change number of packets to send? Default 1. (y/n): y
Packets to send: 5
Origin ID: 0
Destination ID: 15
Verbose mode? (y/n): y

Lost packet! From 6 to 10.
0 -> 1 -> 2 -> 6 -> 10 -> 11 -> 15
Travel time: 24.2483 ms
1 lost packet(s)

0 -> 1 -> 5 -> 10 -> 14 -> 15
Travel time: 16.851 ms
0 lost packet(s)

0 -> 4 -> 8 -> 12 -> 13 -> 14 -> 15
Travel time: 24.1297 ms
0 lost packet(s)

Lost packet! From 2 to 6.
0 -> 1 -> 2 -> 6 -> 10 -> 14 -> 15
Travel time: 24.7303 ms
1 lost packet(s)

0 -> 1 -> 2 -> 3 -> 7 -> 11 -> 15
Travel time: 24.2657 ms
0 lost packet(s)
```

Fig. 6: Program output with five packets from router 0 to 15.

In a normal shortest path simulation, each packet would take the same route each time. However, with the delay-based search, the packets take the path with the lowest amount of delay, and therefore, lowest congestion. In the above example, none of the packets take the shortest path as shown in Figure 2, implying that it was heavily congested.

From this, it can be concluded that this routing algorithm will take a longer path if it means avoiding congestion.