Computer Networks and Applications (COMP3331)

Programming Assignment:

Routing Performance Analysis

Nicholas Ho z3422527

Steven Ru z3418434

# Data Structures

We have chosen to represent the topology with an undirected graph, which is made up of 3 Java classes:

### Node

This represents each of the routers in the network.

Contains 3 fields:

* **String** name: The name of the node (A or B or C etc)
* **Double** minDistance: Keeps track of the minimum distance this particular node is from the start Node. Required for Dijkstra’s Algorithm.
* **Node** prev: Keeps track of the previous Node object that was visited before the current Node was visited.

### Edge

This represents each of the links in the network. Since the graph is an undirected graph, we represent each undirected Edge with 2 directed Edges. Any operations on such an undirected Edge will affect both Edges running in either direction.

Contains 5 fields:

* **Node** from: Contains one of the two Nodes at either end of the Edge.
* **Node** to: Contains one of the two Nodes at either end of the Edge.
* **Int** propagationDelay: Propagation delay is an argument taken in from topology.txt.
* **Double** numSimulCircuits: numSimulCircuits is an argument taken in from topology.txt.
* **ArrayList<VirtualCircuit>** circuits: This is the list of Virtual Circuits that are currently using this particular Edge as a Link.

### UndirectedGraph

This represents the entire network.

Contains 2 fields:

* **ArrayList<Node>** nodes: Contains the list of all nodes or routers that are involved in this network.
* **ArrayList<Edge>** edges: Contains the list of all edges or links that are involved in this network.

# Comparison of Performance Metrics (VCN)

Temporary values. Still need to fix bug regarding bidirectional edges.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Total VC | Total  pkt | Succ. Routed  pkt | % routed  pkt | Num blocked  pkt | % blocked  pkt | Avg hops | Cum. Prop delay |
| SHP | 8377 | 259106 | 231891 | 89.50 | 27215 | 10.50 | 2.37 | 167.48 |
| SDP | 8377 | 259106 | 236516 | 91.28 | 22590 | 8.72 | 3.04 | 140.49 |
| LLP |  |  |  |  |  |  |  |  |

# Analysis of Results

As can be seen, Total VC and Total packets are not affected by the type of algorithm used, and remain constant at 8377 and 259106. However, the number and percentage of successfully routed packets is slightly higher for SDP (236516, 91.28%) than SHP (231891, 89.50%). Correspondingly, the number and percentage of blocked packets for SDP (22590, 8.72%) is lower than that of SHP (27215, 10.50%)

The reason for the lower amount of blocked packets for SDP could be the higher average number of hops (3.04) than SHP (2.37). Because a greater number of links are involved in SDP, there are (3.04 – 2.37 = 0.67) more links to take some of the load, as compared to SHP, where the load is shared between only 2.37 links. This means that there is a lower likelihood that the number of virtual circuits involved in the links in SDP will exceed the numSimulCircuits field in each Edge than in SHP.

Additionally, the cumulative propagation delay for SDP is 142.25, smaller than that of SHP, 173.53. This smaller propagation delay is directly beneficial for SDP, given that the SDP algorithm uses propagation delays of the Links as the length of the path taken. This means that packets take a shorter time to travel the distance from source node to destination, and its resources can be freed for use by other packets more quickly.

Together, the greater average hops yet lower propagation delay for SDP allows it to route more packets successfully, resulting in fewer blocked packets.

# Evaluation of VPN