|  |
| --- |
| COMP3331 |
| Programming Assignment |
| Routing Performance Analysis |
|  |
|  |
| **Thien Nguyen – z3288816**  **Chetan Sahai – z3288822** |

|  |
| --- |
|  |

# Data Structures

## Workload

The workload was stored as a queue of *Requests* in the *Workload* class. After parsing the *workload.txt* input file, a queue of *Requests* is created and ordered by the requested time. Each *Request* contains

* Source node name
* Destination node name
* The Path between the nodes (a list of nodes between source and destination)
* Timestamp and Duration

The path remains uninitialized until the request is processed and a path is calculated using the routing processor.

## Network Topology

The network topology was stored as a graph in the *Network* class. The graph was broken into vertices for each node defined in *topology.txt* and Edges for each connection between nodes defined in *topology.txt*. Vertices were stored in a hash table referenced by their node-name stored as strings. Each vertex then contained a hash table of connected Edges, referenced by the other end of the Edge. Edges contain information on their propagation delay, virtual circuit capacity and number of active connections.

Class *Network*

HashMap<String, Vertex> *Nodes*

“A” “B” “C”

Class *Vertex*

String *name* = “A”

HashMap<String, Edge> *adjacent*

“B”

“C”

Class *Vertex*

String *name* = “B”

HashMap<String, Edge> *adjacent*

“A” “C”

Class *Vertex*

String *name* = “C”

HashMap<String, Edge> *adjacent*

“B”

“A”

Class *Edge*

Class *Edge*

Class *Edge*

PriorityQueue <Request>

*activeConnections*

The network also contains a queue of active connection requests, used to keep track of loading on the network. This data structure is illustrated in Figure 1.

Figure 1 - Diagram of network topology data structure

# Comparison of Routing Protocols

Table 1: Comparison of the three routing protocols \*

|  |  |  |  |
| --- | --- | --- | --- |
| *Performance Metrics* | *SHP* | *SDP* | *LLP* |
| Total number of virtual circuit requests | 8377 | 8377 | 8377 |
| Total number of packets | 259106 | 259106 | 259106 |
| Number of successfully routed packets | 246720 | 246734 | 258616 |
| Percentage of successfully routed packets | 95.22 | 95.23 | 99.81 |
| Number of blocked packets | 12386 | 12372 | 490 |
| Percentage of blocked packets | 4.78 | 4.77 | 0.19 |
| Average number of hops per circuit | 3.67 | 4.27 | 4.96 |
| Average cumulative propagation delay per circuit | 168.45 | 139.84 | 243.43 |
| *\* Virtual circuit network with frequency of 1 packet/sec* | | | |

The LLP routing algorithm provides the highest percentage of successfully routed packets, at the expense of propagation delay and hop count, i.e. more packets will reach their intended destination, but end-to-end latency and hop count will be higher than for the other routing schemes. This is because LLP actively seeks out those paths that are not at capacity so that as few packets as possible are dropped, and this is prioritized over metrics like delay or number of hops. Since avoiding over-capacity links can result in longer paths, the average number of hops and average cumulative propagation delay per circuit are larger than for the other two algorithms.

SDP offers the lowest end-to-end delay of the three algorithms and a lower average hop count than LLP, but this comes at the expense of successful delivery. This is because SDP prioritises those paths that result in the lowest end-to-end delay, and ignores the capacity of each link or the number of hops. As a result, packets can be dropped if the links that would offer the lowest delay are at or near capacity. Similarly, the path that would offer the lowest delay isn’t necessarily the path that offers the fewest hops, so the hop count is higher than for SHP.

SHP meanwhile, offers roughly the same level of success at routing packets as SDP, but with higher latency. The hop count is, as expected, lower than either of the other two algorithms, and this is because hop count is the only metric considered when creating a path. Since the number of hops required to get from point A to point B doesn’t necessarily correspond to the path with the shortest delay, the end-to-end latency is higher than for SDP. Packets can be dropped if the links that would offer the lowest hop count are at or near capacity.

# Virtual Packet Network Performance Evaluation