COMP3331 – Assignment 2 Report

Christopher Michael Phibbs (z3460482)

&

David Sicong Zhou (z3378717)

Data Structures Explanation

The topology of the network was stored in a global 26x26 two-dimensional array (adjacency matrix), due to the fact that there were a maximum of 26 possible different nodes. Each element of the two-dimensional array contains a link struct type, which contains information on the names of the nodes it connects, the amount of propagation delay associated with said link, as well as the current and max loads of that particular link. The link essentially stores each line of Topology.txt.

The links were stored in this fashion so that when each routing protocol algorithm was invoked, it would only have to look at the field of each element in the matrix rather than having to access some other type of data structure containing this information (e.g. an array or linked list), which in turn made all the routing algorithms much more efficient. The adjacency matrix was used as a means of storing these links because each node name effectively represented an integer value from 0 to 25, meaning that it was very easy to refer to the adjacency matrix values based on these node names. This method was much preferred to alternatives such as an array of linked lists, as each element of the array would need to have been iterated through completely when searching for a particular link, in turn reducing the algorithm's efficiency.

Each line of Workload.txt was processed into a request struct. After compiling the list of requests from workload.txt, requests would be broken into packets with lifetimes dependent on either CIRCUIT or PACKET network schemes. Circuit network packets would consume the entire duration, where Packet network packets would be split per (1/packet rate) and routed for each packet.

Each packet would store its original request pointer, its start and end locations, start and end time, and its path. Packets are to be put into a priority queue based on start time, and when read, it will then use the packet's path to refer to the adjacency matrix, to find the links to increase the load of. If the link is maxed out, the packet fails.

Finally, packets store "willDie", as once a packet has begun consuming the load of all links in its path, the packet is reinserted to the priority queue except now based on its end time. When the priority queue reaches a packet that will die, it will then use the packet's path to refer to the adjacency matrix, to find the links it must decrement the load counter for. This assures that each packet validly is processed to whether or not the packet is successfully routed.

In retrospect, the priority queue could have been processed immediately until the start of the next request (to avoid invalidating the network). This would have kept the size of the queue down as new packets were created, rather than build an exponentially large queue.
Instead, the code definitely runs over 4 minutes for the sample file.

Comparison of Performance Metrics

Since not all functionality was implemented, rough estimations are provided based on our inference of the network structure and the algorithms.

	Virtual Circuits	Number of Packets	Successfully Routed Packets	% of Successfully Routed Packets	Number of Blocked Packets	% of Blocked Packets	Average Number of Hops	Average Propagation Delay
Circuit - SHP	8000	247500	200000	80	47500	20	2.5	180.00
Circuit - SDP	8000	247500	210000	84	37500	16	3.5	140.00
Circuit - LLP	8000	247500	230000	92	17500	8	3	200.00

Analysis of Results

SHP

The Shortest Hop Path algorithm had a lower number of successfully routed packets and a higher number of blocked packets. This is because the algorithm always tries to use the routing path with the lowest number of hops needed, regardless of the traffic on the nodes in this path. As a result, because packets for a circuit are continuously sent along this same path, the nodes reach their maximum capacity at a faster rate and thus drop more packets.

On the other hand, the average number of hops for this algorithm is much lower. This is obviously due to the fact that the algorithm finds the route with the least hops required – thus meaning that it will have the lowest average number of hops out of all 3 routing algorithms.

SDP

The Shortest Delay Path algorithm had the lowest average propagation delay of the 3 algorithms. This is because the algorithm selects the route with the lowest total propagation delay across each node in the path. Because this algorithm does not care about how many hops to get from its source to its destination, or the load on each node, it obviously does not achieve the highest successful number of routed packets nor the lowest number of average hops out of the 3 algorithms.

LLP

The Least Loaded Path algorithm had the highest number of successfully routed packets and the lowest number of blocked packets. This is because the algorithm seeks a routing path with the lowest amount of traffic on each node. Because the resulting path is guaranteed to be the path with the least load on each node, it is more likely for each packet to be able to successfully enter each node's buffer queue. As a result, each packet is much more likely to reach its destination without being dropped.

This algorithm also has the greatest average propagation delay, as the least loaded path is typically much longer due to the fact that the path must keep avoiding nodes with more traffic on them, which becomes worse as more circuits are opened simultaneously.

Performance Evaluation of Packet Network

Although completing the code was not accomplished on time, the process of coding has given a good estimate on the results between networks and algorithms.

For circuit networks, the packet rate is irrelevant to how often a path is occupied. Packet networks obvious will occupy a variety of routes in small bursts instead.

SHP will always result in the smallest amount of hops, which becomes more effective the closer each link's delay is to each other. Each packet being routed will produce similar routes, so the difference from a circuit network is not as significant
SDP aims to minimise depth, and similar to SHP is not significantly different from a circuit network.
LLP however has the largest difference. When actively rerouting each packet with a focus on prevent failures, the lowest packet failure percentage out of any network/algorithm combination can be achieved.