COMP3331 Assignment 2: Alex Nguyen (z3379933) and Brian Lam (z5035087)

1. Explanation on the data structures used for the internal representation of the network topology.

- Our program runs on Python3 and is managed by a RoutingPerformance class in routing_performance.py in the root directory. It also only handles the CIRCUIT case.
- RoutingPerformance instantiates the workload queue and network topology graph with the aid of data structures defined in helper/*
- Each requested connection is represented by a VirtualConnection class which contains the desired start/source/destination/duration, a path field to be filled in by a shortest_path algorithm and a fill_path method to call on a path
- In the program, time is simulated by working through a queue of actions with connections, prioritised on (time_of_access, connection). Every time a request is popped off, time passage to the time_of_access value of a connection is simulated. Hence, everytime a request is successful, another work request action is added to free up capacity when the duration has expired. All time has passed once all work requests are processed
- The Graph class in helpers/graph_rep.py maintains state of the network at all time by tracking VirtualConnections currently on, edges, delays, capacities and max_capacities. It parses topology.txt and signals to RoutingPerformance whether a connection will pass or block
- helpers/workload_queue.py holds priority queue data structures for use in shortest_path (helpers/pathing_algorithms.py) as well as the main workload queue. See comments on function
- Routing performance has a StatisticsManager that also tracks and records statistics through counters as it works through the processing queue
- Pathfinding in each case (SHP/SDP/LHP) is handled by modifications of the PriorityQueue version of Djikstra's Shortest path. Passing in a flag for routing protocol will cause a slightly different method of calculation for an updated cost to a specific link (the code refers to this as added/calc and altered delay). Randomosity of path choice is implemented by random number generator with 50% chance of breaking a tied 'cost' path (overwriting the shortest path when it finds one of equivalent cost) during the update of distance to a neighbouring vertex
- Early architecture diagram is also provided in docs/Structure_overview.pdf

2. Comparison of performance metrics for 3 routing protocols over the virtual circuit.

The following results are based off a PACKET RATE = 1.

Performance Metrics	SHP	SDP	LLP
Total number of virtual circuit requests	5884	5884	5884
Total number of packets	176067	176067	176067
Number of successfully routed packets	160292	159927	172278
percentage of successfully routed packets	91.04	90.83	97.85
number of blocked packets	15775	16140	3789
percentage of blocked packets	8.96	9.17	2.15
average number of hops per circuit	3.66	4.31	5.30
average cumulative propagation delay per circuit	166.93	140.21	265.87

3. Analysis of performance results

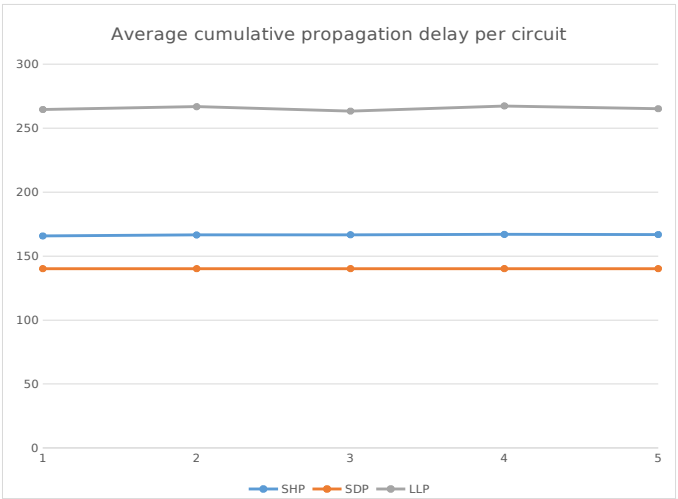
As we can see, the LLP routing protocol resulted in the highest percentage of successfully routed packets followed by SHP then SDP. The arises as the pathfinding in LLP actively considers the prioritises the capacity of paths at each point, reducing the likelihood of blockage and trading off efficiency in path length/propogation delay in the process (delay/hops is not considered). In contrast, both SDP and SHP have no consciousness of other current connections (and hence are blind to likelihood of blocking) which shows in their lower routing success rate. SHP is more naive as only hop number is prioritised (explaining having the lowest avg hops per circuit) which shows in its avg propogation delay being higher than SDP. It is worth noting that SHP is most vulnerable to randomness in tie breakage, as the difference in % success is low between SDP & SHP, with some runs resulting in SDP > SHP. The Total number of virtual circuit requests and packets are identical in each case because the same workload is

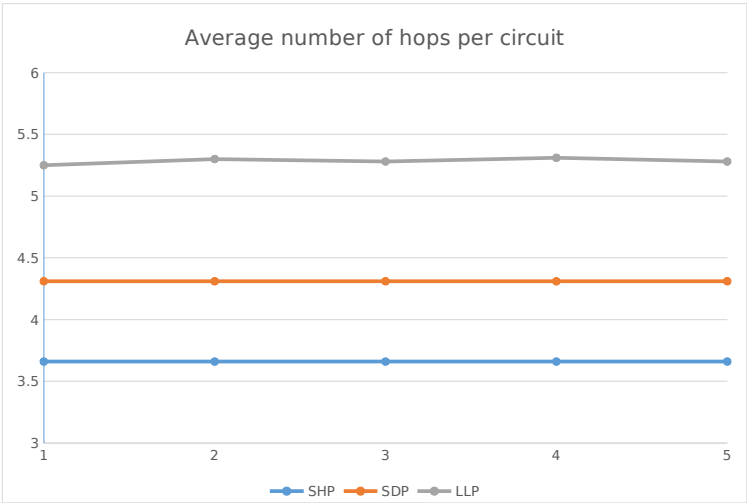
used in all cases. The following discussion with varying packet rates discusses reasoning behind the percentage of blocked packets, average number of hops per circuit and average cumulative propagation delay per circuit for each routing protocol.

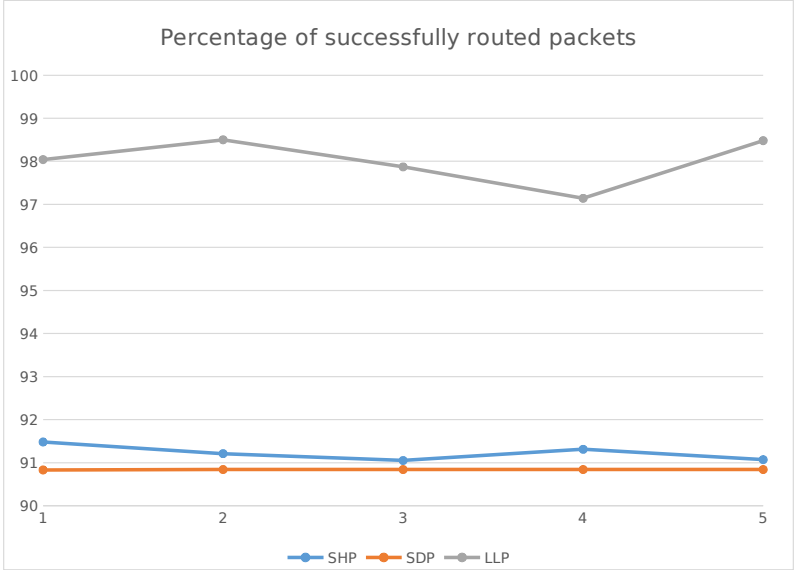
4. Performance evaluation of Virtual Circuit network with varying packet rates + Plots

Varying the packet rate results in the table and plots below, with analysis to follow:

Percentage of successfully routed packets					
	1	2	3	4	5
SHP	91.48	91.21	91.05	91.31	91.07
SDP	90.83	90.84	90.84	90.84	90.84
LLP	98.04	98.5	97.87	97.14	98.48
Average number of hops per circuit					
	1	2	3	4	5
SHP	3.66	3.66	3.66	3.66	3.66
SDP	4.31	4.31	4.31	4.31	4.31
LLP	5.25	5.3	5.28	5.31	5.28
Average cumulative propagation delay per circuit					
	1	2	3	4	5
SHP	165.78	166.56	166.72	166.97	166.85
SDP	140.21	140.21	140.21	140.21	140.21
LLP	264.67	266.86	263.42	267.48	265.28







Analysis on percentage of successfully routed packets.
The Least Loaded Path algorithm is observed to have the highest rate of success when routing packets compared to Shortest Hop Path and Shortest Delay Path, which block roughly 5-6% more packets. The main difference between LLP and the two other algorithms is that it focuses on link load as a cost rather than the number of hops or propagation delay. Because of this focus, link loads (on average) across the network would be less “congested” as the algorithm chooses paths with the least load, resulting in links that stay open for longer thus increasing the rate of success for routing packets. With varying packet rates, routing success with LLP seems to fluctuate across a ~2% band, whereas in the case of SHP and SDP there is very minimal or even no change.

Analysis on average number of hops per circuit.
The Shortest Hop Path has the smallest average number of hops per circuit, followed by the Shortest Delay Path and the Least Loaded Path with the highest average number. Between each algorithm, there is a ~1-2 hop difference proportional to each other. SHP clearly results in the least hops, as the algorithm disregards propagation delay and link load as a cost and instead focuses on reducing the number of hops it takes along the path. SDP comes second, taking into account propagation delay. Inherently, this would mean a trade-off with an increased amount of hops in return for a lower total delay across the path. Lastly, LLP has the highest average number of hops per circuit as it purely addresses lowering link load and completely disregards hops and delay costs. It does not mind taking much longer/delayed paths as long as they have capacity to spare. The varying packet rates have very little to no effect on the average number of hops.

Analysis on average cumulative propagation delay per circuit.
The Shortest Delay Path algorithm has the smallest average cumulative propagation delay per circuit, with Shortest Hop Path following closely by and the Least Loaded Path producing the highest delay figures. The SDP results display the lowest figures, as its algorithm takes into account primarily the propagation delay of each link and tries to generate a path which minimises delay. As a result, it would have the lowest average cumulative delay per circuit. SHP has the next lowest delay per circuit since it focuses on reducing the number of hops along a path. Inherently, this would lead to a lowered average cumulative delay as packets would traverse the smallest number of paths possible. But taking into account that some paths may have an abnormally high delay cost, SHP on average would not fare as well as SDP with finding a path with the least propagation delay. Finally, LLP is observed to have the highest average cumulative propagation delay per circuit, with approximately 100-150ms difference from results generated by the other two algorithms. This is because LLP addresses only link load as a cost, which means that a packet may have to traverse through nodes which aren’t necessarily towards the direction of the destination node in order to minimise link load thus increasing its delay cost. Similar to average number of hops, average delay per circuit has very little relevance to varying packet rates.

5. Link to screencast demo

The screencast can be found here: https://www.youtube.com/watch?v=7mzHfbub0Z8