COMP3331 Assignment

Analysis of circuit switching protocols

George El Boustani
Daniel Morton

Oct 17, 2013

Description of the data structure used to create the internal representation of the network topology

The network topology is represented as a 26 by 26 adjacency matrix, with a column for each node and a row for the edges, found in the class called VirtualNetwork. Information regarding the links (edges) between nodes is represented in an object called LinkInfo, holding information for the delay, the size, the current number of connections and a list of expiry times. For example, graph[0][2] would contain information (in the form of a LinkInfo object) about the link between A and C.

The topology file is used to generate this adjacency matrix. If no link exists between two nodes, then the corresponding index in the matrix would remain a null.

Tabulated comparison of the performance metrics of the 3 routing protocols for the provided topology and workload files

Property	**Shortest Hop Path**	**Shortest Delay Path**	**Least Loaded Path**
total number of virtual circuit requests	5884	5884	5884
number of successfully routed requests	5235	5340	5799
percentage of successfully routed request	88.97	90.75	98.56
number of blocked requests	649	544	85
percentage of blocked requests	11.03	9.25	1.44
average number of hops per circuit	2.36	3.00	3.97
average cumulative propagation delay per circuit	150.30	127.25	245.02

Explanation of the performance results observed:

Shortest Hop Path (SHP)
This algorithm had the highest percentage of blocked requests at **11.03%**. This algorithm does not recognise the capacities available on links in the network, and hence it is understandable that its percentage of blocked requests is higher than the other two algorithms.

SHP, as the name suggests, favours the route with the shortest number of hops. This is evident in its significantly lower average hop count when compared against the other algorithms. This algorithm has an average hop count of 2.36, which is a reasonable value. The only notion of cost SHP places is the number of hops taken, and so we can assume the value of 2.36 is attained as a result of circuit requests for which no direct path exists between the desired nodes, as opposed to other algorithms which purposely desire other paths for delay or congestion avoidance purposes.

This algorithm favours the path with fewest hops, and naturally this leads to a lesser overall average cumulative delay if we are to assume the delays between nodes are randomly spread. In effect we can see this algorithms average cumulative delay to be quite normal, sitting between the two other algorithms at 150.3 milliseconds.

Shortest Delay Path (SDP)
This algorithm was slightly better than SHP in regards to the percentage of blocked requests, with a value of **9.24%** failures. This algorithm does not directly take into account the congestion of certain links, and this is consistent with its relatively higher blockage rate when compared against the LLP algorithm.

Unlike SHP, this algorithm does not consider hops but rather the delay of the hop, when calculating costs. We can see this justifies the slightly increased average hop count of 3, as this algorithm will favour a route which leads to the shortest overall delay time, even if it means making extra hops to achieve the shorter delay.

This algorithm favours the path with the shortest delay, and thus we expect it to have the lowest average cumulative propagation delay. As we can see, its value is 127.25, much lower than the other two algorithms, and especially the LLP's.

Least Loaded Path (LLP)
This algorithm has the best performance when it came to serving requests, where only **1.44%** of requests were blocked. We can see that this is a result of LLP's desire to avoid the more congested links by costing them more, hence leading to a chosen path which has a lower probability of being full. This in turn leads to more successfully fulfilled requests. We also should note that we do not attempt to find another path when a chosen path is later found to contain a blocked link, as requested in the spec.

LLP can be seen to have a significantly higher average hop count when compared to the two other algorithms, taking 3.97 hops on average. Similarly to SDP, LLP does not take hop count into consideration at all, but rather favours the least congested links at all times. This makes it easy to see why LLP's average hop count is higher than SHP's. We can also explain why it is much higher than the SDP by realising that the SDP favours shorter delay times. Clearly if delay between nodes is random, a shorter delay can generally be attained by following a route with less hops. LLP neither considers hop counts or delay times, and therefore does not benefit in the same way the other algorithms do when it comes to average hop count values. A less congested path has no correlation with a smaller number of hops, and in fact may force the search algorithm to favour higher hop paths in order to even out congestion among all links.

This algorithm's main concern is to avoid congested links, and so results in spreading congestion evenly across links in the network. It is therefore reasonable, as we discussed in the hop section above, that paths chosen by this algorithm may tend to have higher hop counts. This couple with the fact that it does not take into account link delay, we see that this algorithms average cumulative propagation delay is much higher than that of the other's, at a value of 245.02 milliseconds.