COMP 9331 Assignment Report

Implementation

I used six threads to implement LSR:

Broadcast thread:
Broadcast their own information when the route goes online. Then the route broadcast every 1 second. When routes need to broadcast, they send the package into sending queue.

Listen thread:
When Routes receiving package from the neighbor routes, put them into receiving queue.

Main thread:
Process different types of package in receiving queue one by one and forward the package by sending them into sending queue.

Sending thread:
Continuously process parcels in the send queue.

Checkalive thread:
I use a dictionary named heartbeats. For example, When receiving package from neighbor A, heartbeats[A]=0. At the same time in checkalive thread, heartbeats[A] add 1 per second. When heartbeats[A] > 3, it means neighbor A is dead. Then this route need to send package to inform other routes about this information.

Dijkstra thread:
Calculate the shortest path between routes by algorithm.

Data structure and link-state packet

Packets have three types. And the form of packet like:
+-----------------------------------+
| TYPE | ID | TIMESTAMP | DATA | Port
+-----------------------------------+

I used three types of packets:

Type 0 means self information broadcast package. I also use it to check dead route.
Type 1 means there is a broken route and the route number is in DATA.
Type 2 means forwards package which include other route's information.

For each route, I use **dictionary named Neighbour_dict** to record its own information. And change **Neighbour_dict into a json** and broadcast it which is DATA in Type 0 packet. When receiving other routes'information, it will updata **a dictionary named Graph** which records the network topology graph.

Graph like { 'A':
{'B':[2.2,5000], 'c':[5.5,5001]}
'B':
{ 'A':[2.2,5000]} }}

Node failure

As I said in checkalive thread before, I use a dictionary named heartbeats. For example, When receiving package from neighbor A, heartbeats[A]=0. At the same time in checkalive thread, heartbeats[A] add 1 per second. When heartbeats[A] > 3, it means neighbor A is dead. Then this route need to send package to inform other routes about this information. And when routes receive type 1 package, they will forward the package to their neighbors. As a result, in each Graph dictionary, the dead note information will be delete.

Restricts excessive link-state broadcasts.

I used a dictionary named restrict_dict to record the time about when this package broadcasting. For example, for route B, restrict_dict[A]=11111.1111s means B last received the package which ID is A (means it contain route A information) , and the broadcast happened in 11111.1111s. So when B received package whichs'Time <= 11111.1111s and ID is A , B will not forward this package.

Design trade-offs considered and made

Before I did this assignment, I did not know how to use multithread to code. So I learn it with my classmates, and there must be some design deficiencies. I did not design an ACK mechanism when route receive the package. Also I just borrowed URL socket code from course web pages.