COMP9331 Assignment2 Report

Outline:

This assignment is mainly about the route protocol of distance of vector, I think the challenges of it are how to design the structure of forwarding tables and how to coordinate with other many threads. Next, I will show how can I achieve these modules and coordinate them.

Demonstration:

Please note that some details when you test this program.
1.This is written in python2.7.

```
$python2.7 DvrPr.py……
```

2.I used 20s as the time interval to judge whether it's stable.Which means if it is in stable, it will also print the output info every 20s.
3.On some special cases, it may need about 3 minutes to converge,please be a little patient.

Part 1: Basic Distance Vector:

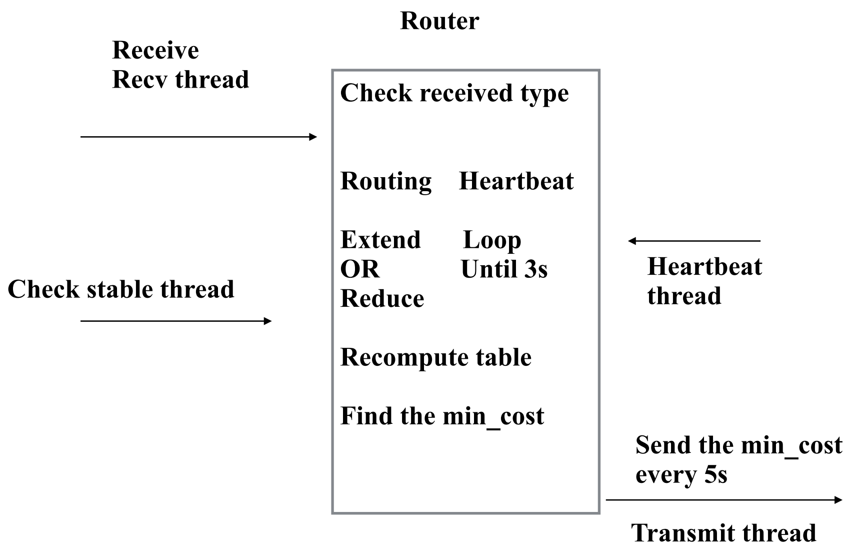
Forwarding Structure:
In order to make my forwarding table more easily to understand. I need to create a list just named marker.
And I will take the node A in the assignment as example to show my ideas.
Form the configA.txt, we can find that there are 3 neighbours, 'B', 'C', 'D', and I just append them into the marker list.
That is markers['B','C','D']
And the format of forwarding table is same as ppt, that is left handside is destinations, and the upper row are the via nodes, let's see the example: table[[2,inf, inf], [inf , 5, inf], [inf, inf, 2]], and I just use each column as a list, note that the markers and table have the same index. Let's take a example, table[0][0] = 2, the makers[0] = 'B', therefore table[0][0] means from node A to B via B the cost is 2.

Extend & Recompute the table:
After the initialising , when one node just say node A received the routing info of B,C,D, the table of A should extend size if necessary and recompute the cost based on the received info. And then find the min cost, send it to neighbours every 5 seconds.

Stable:
How to determine the stable status? I achieved it by comparing the old table and current table between 20s as the interval, which means if the current table is same as table 20s before, that achieve the stable status.

Framework:

I think this graph can help us understand how does the router works.
First, I create receive and send thread to receive and send routing info and heartbeat independently.
Second, check the type of received info, if it is heartbeat, loop it until 3 seconds.Otherwise, extend or reduce the table size based on the received routing info.
Third, recompute the forwarding table. Meanwhile, I create a check stable thread to check whether it's stable every 20s.
Finally, find the min_cost and send them to neighbours every 5s.



Part2: Handling Node Failure

Heartbeat:
I will use the heartbeat to decide the whether the neighbour is alive.Note that every node send its' heartbeat every second as the format, %ID%.And the on the side of receive, when it received data, first should check the type, if it is the heartbeat, just append it to the heartbeat list.And traverse it every 3s, if there are no the neighbour id, just judge that this node is died.

Delete Direct Node:

When the node traverse the heartbeat list and found that there are no neighbour id, then this node would delete this neighbour from the markers and tables.

Reduce table size：
If the node received route info which the number of nodes it has is bigger than local node, traverse them and delete the relevant node.

Part3: Poisoned Reverse

I'd like to take the topology on the assignment as the example to show my ideas.
When the link cost between X,Y changed from 4 to 60,the forwarding table of X,Y should adjust it directly.And the send the route info, infinity to X, or infinity to Y rather than the actual min distance, just as the white lie.The process remaining is just same as before.