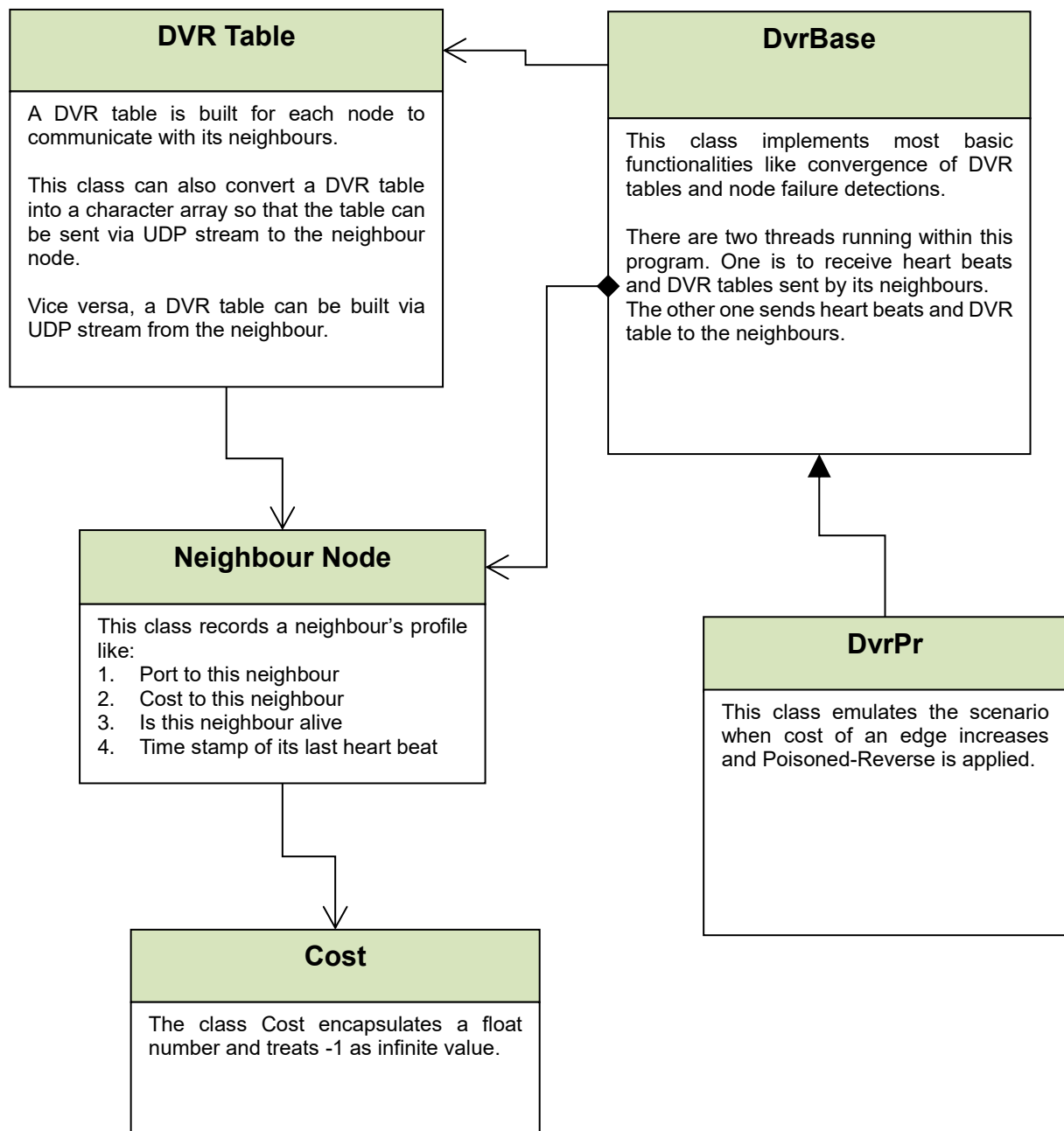# Emulation of Distance Vector Routing

My program of this emulation has implemented the following functionalities.

1. Convergence of DVR tables when the emulated network achieves stable status.
2. Detection of node recoveries and failures of the network
3. Poisoned Reverse mechanism to partly solve the Count-to-Infinity problem when cost of an edge in the network increases tremendously.

Classes Implemented

DVR Table

A DVR table is built for each node to communicate with its neighbours.

This class can also convert a DVR table into a character array so that the table can be sent via UDP stream to the neighbour node.

Vice versa, a DVR table can be built via UDP stream from the neighbour.

DvrBase

This class implements most basic functionalities like convergence of DVR tables and node failure detections.

There are two threads running within this program. One is to receive heart beats and DVR tables sent by its neighbours. The other one sends heart beats and DVR table to the neighbours.

Neighbour Node

This class records a neighbour's profile like:
1. Port to this neighbour
2. Cost to this neighbour
3. Is this neighbour alive
4. Time stamp of its last heart beat

DvrPr

This class emulates the scenario when cost of an edge increases and Poisoned-Reverse is applied.

Cost

The class Cost encapsulates a float number and treats -1 as infinite value.

Program output example

```
weber % java DvrPr C 12002 config_C_12002.txt
-------------------- A is now alive -------------------
Scn DVR
ID: C
Length: 11
        A
A     5.0
-------------------- B is now alive -------------------
Scn DVR
ID: C
Length: 31
        A       B
A     5.0    -1.0
B    -1.0     3.0
-------------------- F is now alive -------------------
Scn DVR
ID: C
Length: 61
        A       B       F
A     5.0    -1.0    -1.0
B    -1.0     3.0    -1.0
F    -1.0    -1.0     5.0
------------- Distance Vector Routing Table changes ------------
Scn DVR
ID: C
Length: 78
        A       B       F
A     5.0    -1.0     9.0
B    -1.0     3.0    10.0
F    -1.0    -1.0     5.0
G    -1.0    -1.0     9.0
-------------------- E is now alive -------------------
Scn DVR
ID: C
Length: 123
        A       B       E       F
A     5.0    -1.0    -1.0     9.0
B    -1.0     3.0    -1.0    10.0
E    -1.0    -1.0     1.0    -1.0
F    -1.0    -1.0    -1.0     5.0
G    -1.0    -1.0    -1.0     9.0
-------------------- D is now alive -------------------
Scn DVR
ID: C
Length: 178
        A       B       D       E       F
A     5.0    -1.0    -1.0    -1.0     9.0
B    -1.0     3.0    -1.0    -1.0    10.0
D    -1.0    -1.0     3.0    -1.0    -1.0
E    -1.0    -1.0    -1.0     1.0    -1.0
F    -1.0    -1.0    -1.0    -1.0     5.0
G    -1.0    -1.0    -1.0    -1.0     9.0
------------- Distance Vector Routing Table changes ------------
Scn DVR
ID: C
Length: 159
        A       B       D       E       F
A     5.0     5.0     4.0     3.0     9.0
B     7.0     3.0     5.0     4.0    10.0
D     6.0     5.0     3.0     2.0     8.0
E     7.0     6.0     4.0     1.0     7.0
F     9.0     8.0     6.0     3.0     5.0
G    13.0    12.0    10.0     7.0     9.0
Shortest path to node A: the next hop is E and the cost is 3.0
Shortest path to node B: the next hop is B and the cost is 3.0
Shortest path to node D: the next hop is E and the cost is 2.0
Shortest path to node E: the next hop is E and the cost is 1.0
Shortest path to node F: the next hop is E and the cost is 3.0
Shortest path to node G: the next hop is E and the cost is 7.0
```

> DVR table grows when new alive neighbours are detected via heart beat signals, or non-neighbours are recognized as alive via neighbours' DVR tables.

> Minus 1 values are assumed to be infinite.

> As the DVR table becomes stable, shortest distances to all other nodes and their first hops are given here.

Problems Unsolved

1. Count-to-Infinity problem cannot be solved by Poisoned Reverse if the edge of increased cost is the only one choice to route to the destination. For example, Poisoned Reverse will not work if the cost between B and C increases.
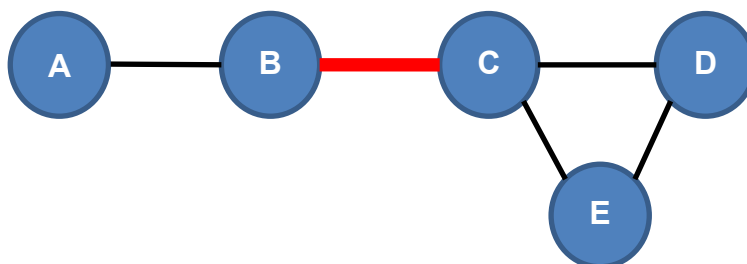
2. There will be Count-to-Infinity problems if the topology of network is partitioned. Though this scenario is NOT required to be handled in this assignment, it is surely an issue.