Report

(i) how distance vector routing works.

Distance vector routing is an asynchronous protocol in order to find the best route from one router to another. It uses the Bellman-Ford equation to find the least-cost path between routers. When a router calculates its Dx(y) for any y in the network it sends this information to all of its neighbors. This is why this algorithm is called for “Routing by rumour” which means that any router in the network determines its forwarding table based on the information it gets from other routers in the network.

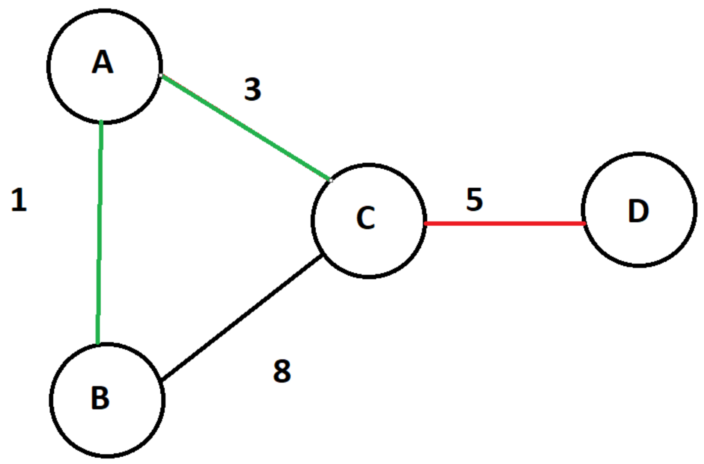
Distance vector is an iterative algorithm meaning that it keeps iterating the processes until no more information is exchanged between neighbors. This makes the distance vector a self-terminating algorithm as well meaning that there is no signal that the computation should stop; it just stops.

(ii) how you tested the algorithms

We tested this algorithm by first: drawing some topologies on the paper in a very simplified version of the network and second: by observing the changes being made after implementing it in RouterNode.py file.

(iii) some cases in which poisoned reverse may fail

Poison reverse can not detect routing-loop in networks that have three or more directly attached neighbors and it still suffers from count-to-infinity problem in these cases.



In the above example poison reverse can not prevent count-to-infinity problems. Imagine when router D fails, then router A will advertise an infinite cost to router D due its route is through router C. Note that the least-cost path to D from A was

through C and that is why poison reverse attempting to address this issue. But we will know here that even A sends a white lie to C but still router B is offering a path to C with a cost of 9 (B to A, A to C and finally C to D). With this new update router C believes that there is a way to D through B. C updates its Distance vector and sends it to router A and B. When A is updated so it affects a more cost for B to get to D and it will loop to infinite as we observe.
We can see even poison reverse can not prevent this issue.

Reference:

- **https://en.wikipedia.org/wiki/Split_horizon_route_advertisement**
- *Computer networking : a top-down approach*

(iv) and a solution to this problem.

There is another solution that can prevent some routing loops described above which is called "Split horizon". The idea of split horizon is that a router never sends an cost-update to the neighbor from which it learned from. In this manner one router never attempts to route to some destination through a router which routes through itself.
This technique helps distance vector protocol to prevent some routing loops in the network.

RIP (Routing Information Protocol) is one of distance vector protocols that implements split horizon technique in order to avoid routing loops even though it may not work all the time. That is when ***hold-down timers*** come into play which makes it even more possible to avoid the formation of routing loops, it happens by ignoring or rejecting the route updates for a while.

Reference:

- https://www.oreilly.com/library/view/cisco-ios-in/156592942X/ch08s03.html
- https://en.wikipedia.org/wiki/Distance-vector_routing_protocol
- https://www.omnisecu.com/cisco-certified-network-associate-ccna/what-is-routing-loop-and-how-to-avoid-routing-loop.php