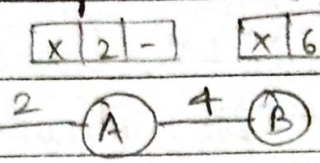
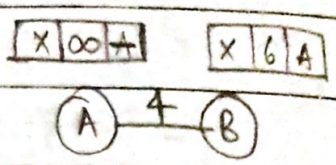
Drawbacks of Distance Vector Routing Protocol:

1. **2-node loop instability.**

(i) Before failure

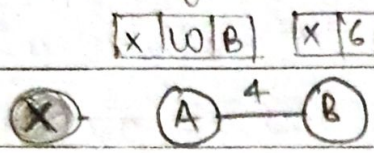


(ii) After failure

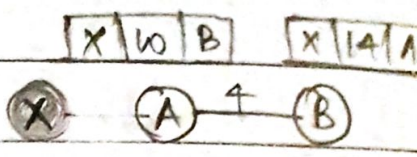


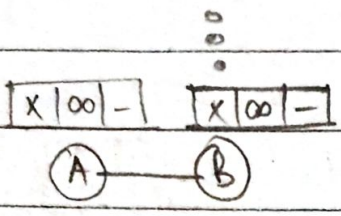
If B sends its routing table before receiving updates from A:

(iii) After A receives update from B



(iv) After B receives update from A





Solutions:

(i) **Defining Infinity** – redefine ∞ to a smaller no., such as 100.
- Most implementations of dist. vector protocol define the dist b/w each node to be 1 & define 16 as infinity.
- However, this implies that DVR can't be used in large systems
- Size of the network in each direction can't exceed 15 hops.

(ii) **Split Horizon** –
- instead of flooding the ~~entire~~ table through each interface, each node only sends part of its table through.
- If, acc. to its table, B thinks that the optimum route to reach X is via A, it does not need to advertise this info. to A.
- The info has come from A (A already knows). Taking the info from A, modifying it, & sending it back to A creates the confusion.

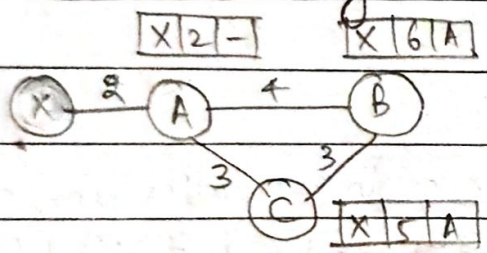
In our scenario, B eliminates the last line of its routing table before it sends it to A. In this case, A keeps ∞ as the dist to X.

(iii) Split Horizon with Poison Reverse –

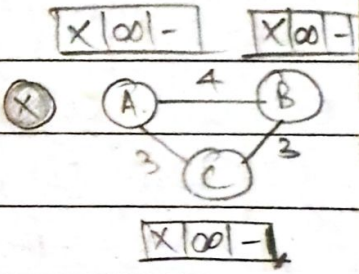
- In this variation of split horizon, when a router sends a routing update to its neighbors, it sends those routes it learned from each neighbor back to that neighbor with infinite cost info. to make sure that the neighbor does not use that route.

2. Three Node Instability

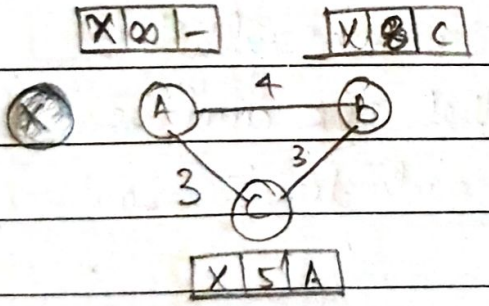
(i) Before failure



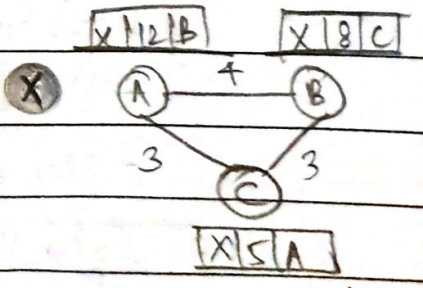
(ii) After A sends the route to B & C, but packet to C is lost



(iii) After B sends the route to A



(iv) After C sends the route to B



- Suppose, after finding that X is not reachable, A sends a packet to B & C to inform them of the situation.
- B immediately updates its table, but the packet to C is lost in the network & never reaches C.
- C remains in the dark thinking that there's a path to X via A with a dist of 5.
- After a while, C sends to B its routing table which includes the path to X. B is totally fooled here.

- It receives the info. on the route to x from C, & acc. to the algorithm, updates its table, showing the route to x with a cost of 8.
- The info. has come from C, not from A, so B may advertise this route to x to A after a while.
- Now A is fooled & updates its table to show that A can reach X via B with a cost of 12.
- The loop continues, & stops when the cost in each node reaches ∞.